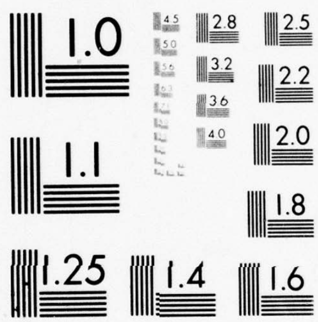
AD-A048 450 ROCKWELL INTERNATIONAL THOUSAND OAKS CALIF SCIENCE --ETC F/G 20/4
A COMPUTATIONAL MODEL FOR THREE-DIMENSIONAL INCOMPRESSIBLE SMAL--ETC(U)
DEC 77 N D MALMUTH, R K SZETO N62269-76-C-0382
UNCLASSIFIED SC5079.5FR NADC-76-105-30 NL

1 OF 2
ADA048450



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NADC 76-105-30

(12)

SC5079.5FR

Copy No. 30

A COMPUTATIONAL MODEL FOR THREE-DIMENSIONAL INCOMPRESSIBLE SMALL CROSS FLOW WALL JETS

FINAL REPORT

For The Period
June 30, 1976 through June 30, 1977

Contract No. N62269-76-C-0382

Prepared for

Naval Air Development Center
Warminster, Pennsylvania 18974

December 15, 1977

Approved for Public Release; Distribution Unlimited

Rockwell International
Science Center

NADC 76-105-30

SC5079.5FR

A COMPUTATIONAL MODEL FOR THREE-DIMENSIONAL INCOMPRESSIBLE SMALL CROSS FLOW WALL JETS

Final Report for the Period
June 30, 1976 through June 30, 1977

Contract No. N62269-76-C-0382

December 15, 1977

Prepared for

Naval Air Development Center
Warminster, Pennsylvania 18974

Norman D Malmuth

Dr. Norman D. Malmuth
Principal Investigator

Norman Malmuth for

R. K. Szeto
Co-Investigator

Unclassified (9) Final rept. 30 Jun 76-30 Jun 77,

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) NADC (19) 76-105-30	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) A Computational Model for Three-Dimensional Incompressible Small Cross Flow Wall Jets.		5. TYPE OF REPORT & PERIOD COVERED Final Report June 30, 1976-June 30, 1977
		6. PERFORMING ORG. REPORT NUMBER (14) SC5079.5FR
7. AUTHOR(s) (10) Norman D. Malmuth and R. K. Szeto		8. CONTRACT OR GRANT NUMBER(s) (15) N62269-76-C-0382
9. PERFORMING ORGANIZATION NAME AND ADDRESS Science Center Rockwell International Thousand Oaks, CA 91360		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (11) 15 Dec 77
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Development Center Code 3014 Warminster, PA 18974		12. REPORT DATE December 15, 1977
		13. NUMBER OF PAGES 106 (12) 108 p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Incompressible flow	Partial differential equations
Viscous flow	Numerical analysis
Jets	Ejectors

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A computational model based on H. Keller's box scheme has been used to characterize turbulent incompressible wall jets in the small cross flow approximation prototypic of flows over upper-surface-blown and augmenter wings. Submerged and coflowing cases are considered. An eddy viscosity model was used to simulate the effects of turbulence. Approximate models are identified for flows in which the jet height tends to zero. If the span flow is introduced through a lateral curvature term appearing in the spanwise momentum equation, the effect of the (OVER)

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified 389 949

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(*When Data Entered*)

20. Abstract (Cont'd)

turbulent coupling on the surface pressures, and peak spanwise velocities is weak.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(*When Data Entered*)

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

ds	differential arc length
s	running arc length along surface, streamwise independent variable
$f(\eta)$	normalized stream function, (p. 22)
h_1, h_2, h_3	metric coefficients
H	jet height
K	constant in logarithmic spiral equation(4.1), (p. 41)
K_1, K_2, K_3	geodesic curvatures, (p. 10)
L	mean radius radius of curvature, (p. 13)
O, o	order of magnitude symbols, (p. 9)
p	pressure
$\tilde{p}$	reduced pressure variable
$\vec{q}$	velocity vector
Q	initial volume flow from slot
R	Reynolds number based on slot height = UH/ν
R_0	initial logarithmic spiral radius
u,v,w	x,y,z components of $\vec{q}$
U	effective mean jet velocity
U_∞	freestream velocity
x,y,z	orthogonal curvilinear coordinates parallel and perpendicular to wall
X,Y	logarithmic spiral Cartesian set
δ	wall jet thickness
Δ	Laplacian, (p. 8)

NOMENCLATURE (Cont'd)

Δ'	Laplacian in x,y
ε	reciprocal of Reynolds number = R^{-1}
$\varepsilon_1, \varepsilon_2$	eddy viscosities
η	Glauert similarity variable, (p. 22)
θ	polar angle
κ	curvature of wall in x,y plane
ρ	density
ψ	stream function, (p. 22)
ω	coflow velocity ratio = U/U_∞ for two-dimensional wall jet
$\vec{\omega}$	vorticity vector
τ	metric function, (p. 8)

<u>Subscripts</u>

o	refers to quantities at jet exit
∞	refers to quantities infinitely far upstream

FOREWORD

This document describes analytical and computational studies of three-dimensional incompressible laminar and turbulent wall jets in small cross flows. This effort was performed during the period June 30, 1976, to June 30, 1977, and was sponsored by the Naval Air Development Center under Contract No. N62269-76-C-0382.

The technical monitor for this study was Dr. K. A. Green.

ABSTRACT

A computational model based on H. Keller's box scheme has been used to characterize turbulent incompressible wall jets in the small cross flow approximation prototypic of flows over upper-surface-blown and augmenter wings with ejectors employing Coanda wall jets. Submerged (i.e., zero secondary flow velocity) and coflowing cases are considered. An eddy viscosity model was used to simulate the effects of turbulence. Approximate models are identified for flows in which the jet height tends to zero. If the span flow is introduced through a lateral curvature term appearing in the spanwise momentum equation, the effect of the turbulent coupling on the surface pressures, and peak spanwise velocities is weak.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation for the valuable comments and review of this report by Drs. K. A. Green, W. D. Murphy, and V. Shankar. Contributions of J. D. Cole relevant to Section 2.5 are also acknowledged.

1.0 INTRODUCTION

Modern naval aircraft can reduce strike force vulnerability by distributing these vehicles over a larger number of ships within the fleet. One way of achieving this allocation is through the attainment of vertical lift-off capability. A technique used to provide vertical lift without oversizing the engine in the cruise mode is the use of thrust augmenting ejectors. With these devices, engine thrust can be enhanced during vertical takeoffs and landings. Obviously, it is desired to achieve the highest thrust augmentation ratio (ϕ) as possible. Various design concepts have been advanced toward obtaining this goal. In the Navy/Rockwell International XFV-12A, for example, an ejector system composed of a centerbody and two Coanda wall jets is currently under development. A central feature of the flow fields produced by this system is three-dimensionality. This has been particularly evident in subscale flow visualization on the Coanda surfaces. It is believed that these flow processes may be important toward ϕ maximization. One way of understanding this relationship is through theoretical modeling which can provide a means of reducing the high cost of powered lift testing. Unfortunately, existing methodology has been limited in the past to two-dimensional flows for the analysis of wall jets and complete ejector systems.

In Ref. 1, a semi-analytical solution for a wall jet over a flat plate is considered. Both the cases of laminar and turbulent flow are treated. Similarity solutions are studied for the laminar case in which the flux of exterior momentum flux is an invariant. For the flat plate case, the existence of this constant does not depend on similarity. With regard to two-dimensional

laminar jets over a curved wall treated in Ref. 2, similarity is necessary to obtain the corresponding invariant. Two-dimensional turbulent wall jets were also considered by Giles, et al.,[3] who studied self-preserving behavior for logarithmic spiral profiles. Various workers have studied turbulent processes experimentally in two- and three-dimensional wall jet flows. This effort is exemplified by Refs. 4-7. Coflowing jets which in contrast to the submerged case have the jet embedded in an external inviscid field are of great practical interest. Kruka and Eskinazi have investigated deviations from similitude in such flows as well as merging of the mixing and wall layers.

Three-dimensional turbulent processes have been studied in connection with downstream behavior of non-circular jets over flat plates and are exemplified by Refs. 9 and 10. These investigations have relevance to the prediction of ejector three-dimensional mixing described in Ref. 11.

The mathematical prediction of these flows presents formidable problems. Only the simplest geometries, e.g., flat plate or special wall shapes such as the logarithmic spiral, lead to an ordinary differential equation for a similarity solution. For turbulent flows, with realistic eddy viscosity models, partial differential equations govern the flow field. Modern finite difference methods offer promise of handling these cases. In particular, Dvorak in Ref. 12 treats two-dimensional wall jets over boundaries of large curvature. Computational modeling of three-dimensional generalizations of these flows has up till now been unexplored to the best of our knowledge. This class of flows occurs in connection with taper and sweep effects on lift augmenters and upper-surface-blown wings.

To shed light on the associated flow patterns, a study, "Three-Dimensional Flow of a Wall Jet," was initiated by the Naval Air Development Center to investigate wall jet flows which exemplify typical features of complex propulsive lift applications. The purpose of this study has been to apply modern computational methods to the treatment of three-dimensional wall jets. The following three basic tasks were performed:

Task 1: Formulate a model to describe a 3-D wall jet in the small cross flow approximation.

Task 2: Develop a numerical method and computer code to treat a 3-D wall jet.

Task 3: Parametric studies using computer code.

In Task 3, the streamwise developments of shear stresses, sideslip angles, streamwise, and spanwise velocity profiles have been studied.

This report will summarize the basic results for all three tasks.

2.0 FORMULATION OF THE PROBLEM

2.1 Description of Physical System and Assumptions

The configuration shown in Fig. 1a has formed the basis of this investigation. Depicted is a section of a three-dimensional wing OPCEFO which has a wall jet over its surface ADEF generated by the efflux from the slot ABCD. An intrinsic coordinate system (x,y,z) is arranged so that the slot ABCD is embedded in the surface $x = 0$, and the wing is the surface $y = 0$. Surfaces x = constant are normal to the wing and orthogonal to y = constant as shown in Fig. 1b. For simplicity, a cylindrical arrangement is shown with the z direction parallel to generators of the cylinder. However, the formulation to be discussed can be applied to more complicated three-dimensional shapes.

2.2 Incompressible Navier-Stokes Equations

To serve as a framework for subsequent developments, the incompressible Navier-Stokes equations are considered in this section.

Denoting an arc element ds, and the orthogonal curvilinear coordinate system given in Fig. 1a, and the metric coefficients h_i, $i = 1,2,3$, ds is given by

$$ds^2 = h_1^2 dx^2 + h_2^2 dy^2 + h_3^2 dz^2$$

If u, v, and w are, respectively, the velocity coordinates in the x, y and z directions, then if $\vec{q} = (u,v,w)$, p = pressure, ρ = density,

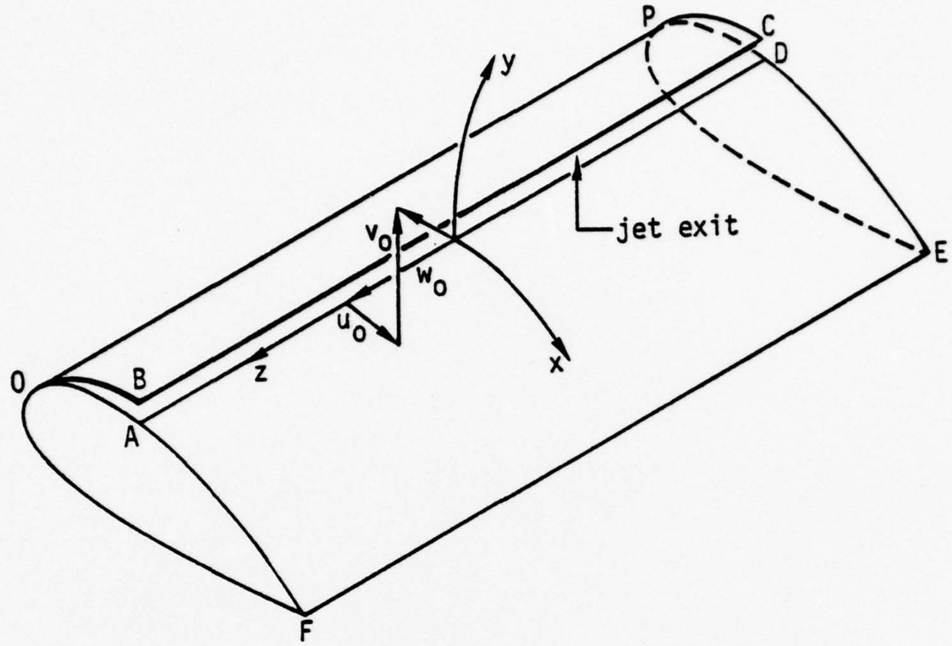


Fig. 1a Geometry of wall jet configuration

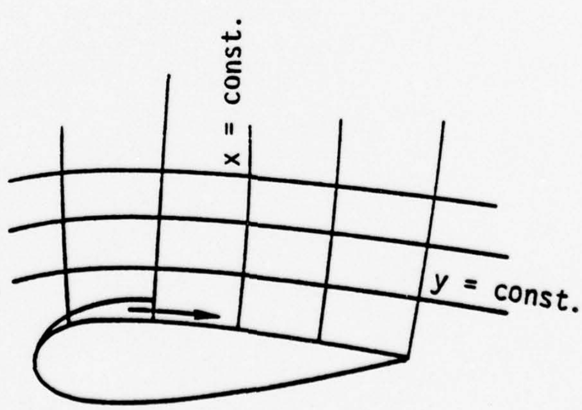


Fig. 1b Intrinsic coordinate system

$\vec{\omega}$ = vorticity = curl $\vec{q}$, then the equations of motion for a laminar flow* with constant kinematic viscosity ν are:

Continuity

$$\operatorname{div} \vec{q} = 0 \tag{2.1}$$

Momentum

$$\vec{q}\times\vec{\omega} = \operatorname{grad}\left(\frac{p}{\rho} + \frac{q^2}{2}\right) - \nu \operatorname{div} \operatorname{grad} \vec{q} \tag{2.2}$$

On taking components, these equations become†

Continuity

$$(h_2h_3u)_x + (h_3h_1v)_y + (h_1h_2w)_z = 0 \tag{2.3a}$$

x Momentum

$$-\frac{v^2h_{2_x}}{h_1h_2} + \frac{uu_x}{h_1} + \frac{v(h_1u)_y}{h_1h_2} + \frac{w}{h_1h_3}(uh_1)_z - \frac{w^2h_{3_x}}{h_1h_3} = \nu\Delta u - \frac{p_x}{h_1\rho} \tag{2.3b}$$

$$\Delta A \equiv \operatorname{div} \operatorname{grad} A \equiv \nabla^2 A = \frac{1}{h_1h_2h_3}\left[\frac{\partial}{\partial x}\left(\frac{h_2h_3}{h_1}\frac{\partial A}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{h_3h_1}{h_2}\frac{\partial A}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{h_1h_2}{h_3}\frac{\partial A}{\partial z}\right)\right]$$

---

*Turbulent flows will be considered in Section 2.6
†Coordinate variable subscripts indicate partial differentiation with respect to these variables.

y Momentum

$$-\frac{w^2}{h_2h_3}h_{3_y} + \frac{vv_y}{h_2} + \frac{w}{h_2h_3}(h_2v)_z + \frac{u}{h_1h_2}(h_2v)_x - \frac{u^2}{h_1h_2}h_{1_y} = \nu\Delta v - \frac{p_y}{h_2\rho}$$

(2.3c)

z Momentum

$$-\frac{u^2}{h_3h_1}h_{1_z} + \frac{ww_z}{h_3} + \frac{u}{h_3h_1}(h_3w)_x + \frac{v}{h_2h_3}(h_3w)_y - \frac{v^2}{h_3h_2}h_{2_z} = \nu\Delta w - \frac{p_z}{h_3\rho}$$

(2.3d)

2.3 Small Cross Flow Approximation

Assuming that w, ∂/∂z << 1, Eqs. (2.3) can be simplified to

Continuity

$$(h_2h_3u)_x + (h_3h_1v)_y = 0$$

x Momentum

$$-\frac{v^2h_{2_x}}{h_1h_2} + \frac{uu_x}{h_1} + \frac{v(h_1u)_y}{h_1h_2} = \nu\Delta' u - \frac{p_x}{h_1\rho}$$

y Momentum

$$\frac{u(vh_2)_x}{h_1h_2} + \frac{vv_y}{h_2} - \frac{u^2}{h_1h_2} h_{1_y} = \nu\Delta' v - \frac{p_y}{h_2\rho}$$

z Momentum

$$-\frac{u^2 h_{1_z}}{h_3h_1} + \frac{u}{h_3h_1}(h_3w)_x + \frac{v}{h_2h_3}(h_3w)_y - \frac{v^2}{h_3h_1} h_{2_z} = \nu\Delta' w - \frac{p_z}{h_3\rho}$$

$$\Delta' \equiv \frac{1}{h_1h_2h_3}\left[\frac{\partial}{\partial x}\left(\frac{h_2h_3}{h_1}\frac{\partial}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{h_3h_1}{h_2}\frac{\partial}{\partial y}\right)\right]$$

2.4 Small Cross Flow, Wall Jet Approximation and Order of Magnitude Analysis

Without undue loss of generality, we consider the case for which*,†

$$h_1 = 1 + \kappa y \quad , \quad h_2 = 1 \quad , \quad h_3 = 1 + \tau(x,z)y$$

$$\Delta = \frac{1}{h_1h_3}\left[\frac{\partial}{\partial x}\left(\frac{h_3}{h_1}\frac{\partial}{\partial x}\right) + \frac{\partial}{\partial y}\left(h_3h_1\frac{\partial}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{h_1}{h_3}\frac{\partial}{\partial z}\right)\right]$$

$$\Delta' = \frac{1}{h_1h_3}\left[\frac{\partial}{\partial x}\left(\frac{h_3}{h_1}\frac{\partial}{\partial x}\right) + \frac{\partial}{\partial y}\left(h_3h_1\frac{\partial}{\partial y}\right)\right]$$

*This notation varies in boundary layer analyses. Some authors prefer $(x,y,z) \to (h_1,1,h_2)$.

†More general h_i's will be considered in future studies.

We consider a wall jet limit involving the jet exit height H becoming small in comparison to the wall radius of curvature at a fixed downstream station x. The jet height δ is $\mathcal{O}(H)$ as $H \rightarrow 0$.* In addition, y, the normal coordinate to the surface is $\mathcal{O}(\delta)$ in the limit. Furthermore, we assume that metric coefficients h_i and K_i, $(i = 1,2,3)$ defined below are $\mathcal{O}(1)$. In this limit, the approximate orders of magnitude of the various terms are shown above the equations tabulated below:

Continuity

$$\underset{\displaystyle (h_3 u)_x}{u} + \underset{\displaystyle (h_3 h_1 v)_y}{v\delta^{-1}} = 0 \qquad (2.4a)$$

(For both terms to balance, v therefore $\sim \delta u$)

x Momentum

$$\underset{\displaystyle \frac{uu_x}{h_1}}{u^2} + \underset{\displaystyle \frac{v}{h_1}(h_1 u)_y}{\delta u^2 \delta^{-1}} = -\frac{p_x}{\rho h_1} + \underset{\displaystyle \nu \Delta' u}{\nu u \delta^{-2}} \qquad (2.4b)$$

(where conclusions from the continuity equation have been used in the ordering)

---

*Considering two arbitrary functions f(x) and g(x), $f = \mathcal{O}(g)$ as $x \rightarrow x_o$ implies that $|f/g| < k$ as $x \quad x_o$ where k is independent of x. The statement $f = 0(g)$ implies that $f/g \rightarrow 0$ as $x \rightarrow 0$.

y Momentum

$$u^2\delta \quad \delta u^2 \quad u^2 \qquad\qquad \nu\delta u\delta^{-2}$$

$$\frac{uv_x}{h_1} + vv_y - \frac{\kappa u^2}{h_1} = -\frac{p_y}{\rho} + \nu\Delta' v \tag{2.4c}$$

z Momentum

$$u^2 \quad uw \quad uw \qquad \delta uw\delta^{-1} \quad \delta^2u^2 \qquad\qquad \nu w\delta^{-2}$$

$$K_2u^2 + K_1uw + \frac{uw_x}{h_1} + \frac{v(h_3w)_y}{h_3} + K_3v^2 = -\frac{p_z}{h_3} + \nu\Delta' w \tag{2.4d}$$

where

$$K_1 = h_{3_x}/h_1h_3 \quad , \quad -K_2 \equiv h_{1_z}/h_1h_3 \quad , \quad \frac{\kappa}{1+\kappa y} \equiv h_{1_y}/h_1$$

$$-K_3 \equiv h_{2_z}/h_3h_1$$

2.5 Finite Momentum Limit for Finite Curvature Walls, ($\kappa = O(1)$)*-- Submerged Wall Jets

To further simplify the foregoing equations, we consider the limit in which $\rho u^2\delta$ is fixed as $\delta \to 0$. Here, $u = O(U)$, where U is defined as a mean jet exit velocity. Accordingly, $u = O(\delta^{-1/2})$. If $\kappa = O(1)$ as $\delta \to 0$, $h_1 \doteq 1$, allowing various terms to be eliminated from the foregoing equations. The ground rules for this process are that at least the frictional term in the

*More complex forms of the equations arise for $\kappa\delta = O(1)$ but will not be considered in this report.

streamwise momentum equation is retained, and a nontrivial y momentum is desired where the pressure gradient normal to the streamlines balances the centrifugal force. If these guidelines are adopted, the approximate equations become, noting that $\partial/\partial z = o(\partial/\partial y)$, and $\partial/\partial y = O(\delta^{-1})$:

Continuity

$$u_x + v_y = 0 \tag{2.5a}$$

x Momentum

$$uu_x + vu_y = \nu u_{yy} \tag{2.5b}$$

y Momentum

$$\rho \kappa u^2 = p_y \tag{2.5c}$$

z Momentum*

$$\frac{uw_x}{h_1} + vw_y + K_1 uw + K_2 u^2 = \nu w_{yy} \tag{2.5d}$$

*The K_1 term is negligible for $h_3 = 1 + \tau(x,z)y$, but is retained here and in Section 2.8 for more general h_3's.

Here, the most interesting case has been retained so that the K_2u^2 term balances other terms in the $\mathcal{O}(uw)$ z momentum equation since it is of order $u^2 h_{1_z}$ which itself is assumed to be $\mathcal{O}(uw)$. For submerged wall jets with $u \to 0$, as $y \to \infty$, in contrast to curved wall boundary layers, the pressure gradient term $p_x/\rho h_1$ is negligible in (2.4b), since from (2.4c), $p = \mathcal{O}(\kappa u^2 \delta) = \mathcal{O}(1)$. A similar result is obtained in the finite mass limit $\rho u \delta$ = fixed as $\delta \to 0$. Only for boundary layers or subregions of coflowing wall jet flows with $\lim_{y\to\infty} u_x \neq 0$ jets implying $p_x = \mathcal{O}(u^2)$ in (2.4b) can the streamwise pressure gradient become important. For the finite momentum limit, inclusion of the friction term in (2.4b) implies $\delta \sim \nu^{2/3}$ as $\nu \to 0$. This order of magnitude has been tacitly assumed in the omission of the higher order term νv_{yy} in (2.5c).

The rationale for the δ scaling with ν and the disappearance of the p_x term from the x momentum equation for submerged jets can be more fully understood from three-dimensional generalizations of asymptotic developments to be discussed shortly in connection with two-dimensional flows. Prior to this, we note that for finite mass with $\rho u \delta$ fixed, $\delta \sim \nu$ as $\nu \to 0$. As will be indicated, other "distinguished limits" are possible in which internal structures such as the wall layer, potential core, and mixing layers can be abstracted.

We conclude this section by noting that the foregoing approximate forms of the equations of motion could be obtained from a formal asymptotic expansion procedure which will be illustrated for two-dimensional curved wall-jets. It is well known that these flows can be divided into a transitional region near the jet exit consisting of a mixing layer, inviscid constant velocity potential

core, and a boundary layer in the vicinity of the wall. The potential core is eaten up by the boundary and mixing layers. Turbulent diffusion results in a merger of these layers at a downstream location. In what follows, we consider the flow in the fully merged zone. As in boundary layers, two different representations can be used to describe the flow structure. An "inner" representation is appropriate to the viscous jet layer near the wall, and an "outer" expansion describes the external inviscid flow. Another option is to develop a uniformly valid asymptotic representation using an optimal set of coordinates developed by Kaplun.[13,14]

Denoting the mean velocity at the exit by $U = Q/\rho H$, where H is the exit height, and Q is the exit mass flow, the exit momentum is QU. Accordingly, the nondimensional form of Eqs. (2.4) in two dimensions can be obtained by normalizing all velocities with respect to U, the pressure difference from ambient with respect to ρU^2, and all lengths with respect to L a mean radius of curvature.* The resulting dimensionless equations of motion are similar in form to (2.4) except with suitable dimensionless redefinitions of the K_1, ρ and the ν coefficients replaced by R^{-1}, where R = Reynolds number based on $L = UL/\nu$.

We now consider appropriate asymptotic representations for the inner viscous layer. Introducing a small parameter ε which is the reciprocal of the Reynolds number R, we envision a sequence of flows observed at a fixed x station in which the normalized wall height, H, is allowed to become vanishingly small as $\varepsilon \to 0$. If $\delta(x;\varepsilon)$ is the characteristic jet height, δ

*Note that other normalizing lengths are possible such as the viscous length ν/U or the jet height H. The velocities can also be referred to a free stream velocity U_∞ provided the latter is not zero. The selection mode here is advantageous for the arguments that follow.

will scale like H as $H \to 0$. To keep the fine structure of the jet layer in view as $H \to 0$, we "blow up" the y scale by a factor $\sigma(\varepsilon)$, where the functional form $\sigma(\varepsilon)$ is to be determined. Here, $\sigma \sim \delta$. To formalize this, we assert that the y dependence is really a dependence on the strained variable $\tilde{y} \equiv y/\sigma$. The most general form of the inner expansion leading to the non-dimensional, laminar two-dimensional specialization of Eqs. (2.5) is

$$u(x,y;\varepsilon) = \varepsilon\sigma^{-2}u_o(x,\tilde{y}) + \varepsilon\sigma^{-1}u_1 + \ldots \tag{2.6a}$$

$$v(x,y;\varepsilon) = \varepsilon\sigma^{-1}v_o(x,\tilde{y}) + \varepsilon v_1 + \ldots \tag{2.6b}$$

$$p(x,y;\varepsilon) = \varepsilon^2\sigma^{-4}p_o(x) + \varepsilon^2\sigma^{-3}p_1(x,\tilde{y}) + \ldots \tag{2.6c}$$

for an "inner limit," $x,\tilde{y}$ fixed as $\varepsilon \downarrow 0$. The "gauge function," σ is determined by matching this solution with the outer inviscid flow.

It should be recognized that for the flat plate boundary layer, since there is no characteristic length in the streamwise direction, the appropriate representations are <u>coordinate</u> expansions for large x rather than for small values of the parameter ε of (2.6). Another viewpoint, see, for example, Van Dyke,[16] is to introduce a fictitious normalizing length in the streamwise direction which cancels out in the analysis.

For wall jets, several "distinguished limits" are relevant for the co-flow ratio $\omega \equiv U/U_\infty$, where U_∞ is the freestream velocity in the outer flow. These cases are as follows:

(i) $\omega \to 0$

(ii) ω fixed

(iii) $\omega \to \infty$

(iv) $\omega = \infty$

as $\varepsilon \to 0$. Case (i) is not of interest for propulsive lift applications. Note further that Case (ii) subsumes Case (iv) which corresponds to a submerged jet. If Case (ii) is assumed, then the assertion that $u = \mathcal{O}(1)$, uniformly in $0 \leq x \leq \infty$, is plausible based on normalization of this streamwise velocity component with respect to U and matching considerations. Accordingly, $\varepsilon\sigma^{-2} = 1$ in (2.6a) implying that $\sigma = \sqrt{\varepsilon}$. This scaling is also appropriate to conventional boundary layer flows. Substitution of (2.6) into the exact equations and retaining terms of dominant order will give the non-dimensional analog of (2.5a)-(2.5c), for the approximate quantities in (2.6), with an additional pressure gradient term in the axial momentum equation due to the coflow effect. These equations are:

<u>Continuity</u>

$$u_{o_x} + v_{o_{\tilde{y}}} = 0$$

<u>x Momentum</u>

$$u_o u_{o_x} + v_o u_{o_{\tilde{y}}} = -p_o'(x) + u_{o_{\tilde{y}\tilde{y}}}$$

<u>y Momentum</u>

$$\kappa u_o^2 = p_{1_{\tilde{y}}}$$

The longitudinal gradient $p_o'(x)$ is determined as in conventional boundary layers by matching with the outer flow streamwise pressure gradient which is determined from Bernoulli's equation. Note that the y pressure gradient balancing centrifugal force across the streamlines arises from the second order term p_1 in the pressure expansion (2.6c).

The representation of the outer flow field is obtained from other asymptotic expansions of the flow variables. The appropriate outer variable normal to the body surface is y and the expansions are:

$$u(x,y;\varepsilon) = U_o(x,y) + \sqrt{\varepsilon}\, U_1(x,y) + \ldots$$

$$v(x,y;\varepsilon) = V_o(x,y) + \sqrt{\varepsilon}\, V_1(x,y) + \ldots$$

$$p(x,y;\varepsilon) = P_o(x,y) + \sqrt{\varepsilon}\, P_1(x,y) + \ldots$$

for x,y fixed as $\varepsilon \to 0$ ("outer limit").

On substitution of these expansions into the exact equations and retaining the dominant terms, the following equations are obtained for the first order quantities:

Continuity

$$U_{o_x} + (h_1 V_o)_y = 0$$

x Momentum

$$U_o U_{o_x} + h_1 V_o U_{o_y} + \kappa U_o V_o = -P_{o_y}$$

y Momentum

$$U_o V_{o_x} + h_1 V_o V_{o_y} - \kappa U_o^2 = -h_1 P_{o_y}$$

To determine the longitudinal pressure gradient in the inner equation and $\sigma(\varepsilon)$, Bernoulli's equation

$$p + \frac{u^2 + v^2}{2} = \omega^{-2}$$

and a matching procedure is used in which the inner and outer solutions are written in a representation appropriate to an intermediate "overlap" domain between inner and outer regions in which the solutions have common validity. For this purpose, the intermediate limit, y_η fixed as $\varepsilon \to 0$, is used in which $y_\eta = y/\eta(\varepsilon)$ and the order of $\eta(\varepsilon)$ is between $\sqrt{\varepsilon}$ and unity. The inner and outer expansions are written in terms of y_η and are equated to various orders, yielding conditions on the unknown quantities.*

From the Bernoulli equation and this procedure, the following boundary conditions are obtained

$$V_o(x,0) = 0 \qquad \text{on} \quad -\infty \leqslant x \leqslant \infty$$

$$P_o(x,0) = p_o(x) = \omega^{-2} - U_o^2(x,0)/2$$

*Van Dyke in Ref. 14 uses Lagerstrom's restricted matching principle to obtain similar results for boundary layers without the intermediate variable formalism applied in this section. Cole in Ref. 17 has applied the intermediate variable matching method for a wide class of singular perturbation problems and has derived formulations similar to those described here for boundary layers.

$$U_o(x,0) = u_o(x,\infty) \equiv u_e(x)$$

$$p_1(x,\tilde{y}) = \kappa\tilde{y}u_e^2(x) - 2u_e(x)U_1(x,0) \quad \text{as} \quad \tilde{y} \to \infty$$

$$V_1(x,0) = \delta^{*\prime}(x) \quad , \quad \delta^* \equiv \int_0^\infty [u_o - u_e]d\tilde{y}$$

The solution procedure is to solve the outer equations with the first of the above boundary conditions. The quantity u_e is subsequently used with p_o to solve the inner problem with an initial condition of the form

$$u_o(0,\tilde{y}) = g(\tilde{y})$$

where g is a prescribed function. In this respect, and the turbulence models employed, the wall jet problem differs from the boundary layer formulation. The latter derives its initial conditions from matching with the outer flow, whereas for wall jets, these are specified independently.

The remaining boundary conditions comprise the no-slip conditions, $u_i(x,0) = v_i(x,0) = 0$ for all i, and the outer boundary conditions for p_1 appearing in the inner y momentum equation.

Note that the quantity $U_1(x,0)$ must be obtained from the solution of the second order outer problem with $V_1(x,0)$ expressed in terms of the slope of the displacement thickness $\delta^{*\prime}(x)$. Higher approximations are obtained using a similar iteration procedure relevant to this weak viscous interaction problem.

Aside from the differences noted, the foregoing problem strongly resembles a boundary layer formulation, for Cases (ii) and (iv). The latter case is obtained from the former by letting $u_e = P_i = U_i = V_i = 0$ for all i. The longitudinal pressure gradient term in the inner x momentum equation is thereby eliminated.

If the velocities are normalized with respect to U_∞ for Case (iii), scalings for the inner variables are obtained which correspond to those derived in the dimensional formulation given in Section 2.5. Formally, the inner equations become in this case

$$u = \varepsilon^{-1/3} u_o(x,\tilde{y}) + \varepsilon^{1/3} u_1 + \ldots$$

$$v = \varepsilon^{1/3} v_o + \varepsilon v_1 + \ldots$$

$$p = \varepsilon^{-2/3} p_o(x) + p_1(x,\tilde{y}) + \varepsilon^{2/3} p_2(x,\tilde{y}) + \ldots$$

for $x, \tilde{y} = y/\varepsilon^{2/3}$ fixed as $\varepsilon \rightarrow 0$. This will yield identical inner equations to dominant order as for Cases (ii) and (iv). However, it is anticipated that details of the matching will be different. As a check, Bickley's similarity solution for a free jet with transverse momentum flux invariant along the jet exhibits the same ε scaling shown in the dominant terms of the foregoing expansions.

It is noteworthy that the submerged jet of Case (iv) is degenerate with respect to (iii). This is plausible since normalizations of the latter are non-existent for $U_\infty = 0$.

2.6 Turbulence Assumptions--Eddy Viscosity Models

In previous studies conducted at the Science Center, a number of turbulence models were investigated. Because of the orientation of this investigation to algorithm development, a detailed study of the adequacy of these models was not attempted. However, it should be noted that the numerical algorithm to be described in subsequent sections is general enough to assimilate the various turbulent models which for the purpose of the present investigation have been restricted to eddy viscosity simulations. For purposes of discussion of the numerical algorithm and the results, the turbulent framework corresponding to Eqs. (2.5) differs inthe respect that the terms νu_{yy} and νw_{yy} in the laminar formulation are replaced, respectively, by their eddy viscosity counterparts $((\nu+\varepsilon_1)u_y)_y$ and $((\nu+\varepsilon_2)w_y)_y$.

A prototypic model selected to illustrate the application of a typical eddy viscosity simulation is:

$$\varepsilon_1 = \varepsilon_2 = \begin{cases} (0.435y)^2 \left[\left|\frac{\partial u}{\partial y}\right|^2 + \left|\frac{\partial w}{\partial y}\right|^2\right]^{1/2} , \quad y < y^* & (2.7a) \\ (0.125y_1)^2 \left[\left|\frac{\partial u}{\partial y}\right|^2 + \left|\frac{\partial w}{\partial y}\right|^2\right]^{1/2} , \quad y \geqslant y^* & (2.7b) \end{cases}$$

where y_1 is determined by

$$u(y_1) = 0.01$$

$$u_y(y_1) < 0$$

$$y^* = \frac{.125}{.435} y$$

It should be noted that this model provides coupling between the spanwise flow w and the streamwise field u not occurring in the weak cross flow

laminar formulation. From a computational viewpoint, the coupling was suppressed to achieve an efficient algorithm. In our procedure, the finite difference approximation used for (2.7) is such that the discretized momentum equations are effectively decoupled. This was achieved by evaluating $\partial w/\partial y$ at the previous streamwise station instead of evaluating the average between the present and last computed streamwise station.

Other turbulence models have been proposed for two-dimensional wall jets in which the wall curvature affects the entrainment and eddy viscosity simulation. These can be accommodated by our computational procedure.

2.7 Boundary and Initial Conditions

The boundary conditions to be employed are the no-slip conditions at the wall and asymptotic conditions relevant to an "outer" flow field external to the jet. Thus, on the wall $y = 0$,

$$u(x,0,z) = v(x,0,z) = w(x,0,z) = 0 \tag{2.8a}$$

At the edge of the jet, $y = \infty$

$$u(x,\infty,z) = u_e(x,z) \tag{2.8b}$$

$$v(x,\infty,z) = v_e(x,z) \tag{2.8c}$$

$$w(x,\infty,z) = w_e(x,z) \tag{2.8d}$$

$$p(x,\infty,z) = p_e(x,z) \tag{2.8e}$$

The initial profile must satisfy the compatibility conditions, i.e., a solution of (2.5) or its turbulent counterpart evaluated at x = 0, subject to the appropriate specialization of (2.8). It should be noted that in an incompressible context, the quantity p_e can be determined from Bernoulli's theorem providing the outer flow is inviscid.

2.8 Formulation in Glauert Variables

To minimize sharp gradients and smooth the computational problem, the governing equations of motion are rewritten in a new set of independent and dependent variables. The Glauert wall-jet transformations given in Ref. 1 are used to change the independent variables (x,y) to $(s,\eta)^*$:

$$ds = h_1 dx \quad , \tag{2.9a}$$

$$\eta = \frac{1}{4} s^{-3/4} y \quad . \tag{2.9b}$$

A new dependent variable $f(s,\eta)$ is introduced such that

$$\psi = s^{1/4} h_2 f(s,\eta) \quad , \tag{2.9c}$$

where ψ is the stream function satisfying the continuity equation with

*The quantities (u,v,x,y,ψ) are dimensionless, being obtained from the corresponding dimensional variables $(\bar{u},\bar{v},\bar{x},\bar{y},\psi)$ by writing $\bar{u} = Uu$, $\bar{v} = Uv$, $\bar{x} = \nu x/U$, $\bar{y} = \nu y/U$, $\bar{\psi} = \nu^2\psi/U$.

$$\psi_y = h_2 u = h_2 s^{-1/2} f_\eta / 4$$

$$\psi_x = -h_1 h_2 v = h_1 (s^{1/4} h_2 f_s + s^{-3/4} h_2 f/4)$$

Furthermore, let $\tilde{p}$ be the reduced pressure given by

$$\tilde{p} = 4s^{1/4} p/\rho \tag{2.9d}$$

Using these transformations, the equations of motion (2.5) simplify to:

Streamwise Momentum Equation

$$\left((1 + \varepsilon_1) f_{\eta\eta}\right)_\eta + \left(1 + \frac{s}{h_2}\frac{\partial h_2}{\partial s}\right) f f_{\eta\eta} + 2f_\eta^2 = 4s(f_\eta f_{\eta s} - f_s f_{\eta\eta}) \tag{2.10a}$$

Vertical Momentum Equation

$$\tilde{p}_\eta = \kappa f_\eta^2 \tag{2.10b}$$

Spanwise Momentum Equation

$$\left((1 + \varepsilon_2) w_\eta\right)_\eta + \left(1 + \frac{s}{h_2}\frac{\partial h_2}{\partial s}\right) f w_\eta + 4K_1 f_\eta w + \sqrt{s}\, K_2 f_\eta^2 = 4s(f_\eta w_s - f_s w_\eta) \tag{2.10c}$$

In addition, the boundary conditions are given by

Boundary Conditions at the Wall, $\eta = 0$

$$f(s,0) = f_\eta(s,0) = w(s,0) = 0 \quad , \quad s \geqslant 0 \tag{2.11a}$$

Boundary Condition at Jet Edge

$$\left.\begin{aligned} f_\eta(s,\infty) &= \sqrt{s}\, R(s) \\ w(s,\infty) &= W(s) \\ \tilde{p}(s,\infty) &= \tilde{P}(s) \end{aligned}\right\} \qquad (2.11b)$$

where R and W are arbitrary functions of s obtained from the external flow, and $\tilde{P}$ can be obtained from Bernoulli's theorem. Consistent with the small cross flow approximation, the dependence on z is absent. It is tacitly assumed in the streamwise momentum equation that $\tilde{P}'(s) << 1$, otherwise, the equivalent of the $-p_x/\rho$ term should be added to the right-hand side of the streamwise momentum equation in accord with a three-dimensional qualitative extension of the matching procedures elucidated in Section 2.5.

3.0 NUMERICAL METHODS

3.1 The Box Scheme

To solve the wall-jet equations in Section 2.7, an implicit finite difference method (the Box Scheme) developed by H. B. Keller[18] is used. The differential equations are written as a first order system in terms of relabeled dependent variables $u(s,\eta)$, $v(s,\eta)$, $t(s,\eta)$:

$$f_\eta = u \tag{3.1a}$$

$$u_\eta = v \tag{3.1b}$$

$$w_\eta = t \tag{3.1c}$$

$$\left((1+\varepsilon_1)v\right)_\eta = -\left(1 + \frac{s}{h_2}\frac{\partial h_2}{\partial s}\right) fv - 2u^2 + 4s(uu_s - f_s v) \tag{3.1d}$$

$$\left((1+\varepsilon_2)t\right)_\eta = -\left(1 + \frac{s}{h_2}\frac{\partial h_2}{\partial s}\right) ft - 4K_1 uw - \sqrt{s}\, K_2 u^2 + 4s(uw_s - f_s t) \tag{3.1e}$$

$$\tilde{p}_\eta = \kappa u^2 \tag{3.1f}$$

(In passing, we note that the right-hand side of the above system (3.1) does not involve terms that are derivatives of η.)

Now consider any family of meshes $\left\{k_n\right\}_{n=1}^{N}$, $\left\{h_j\right\}_{j=1}^{J}$. From Fig. 2 they satisfy the following

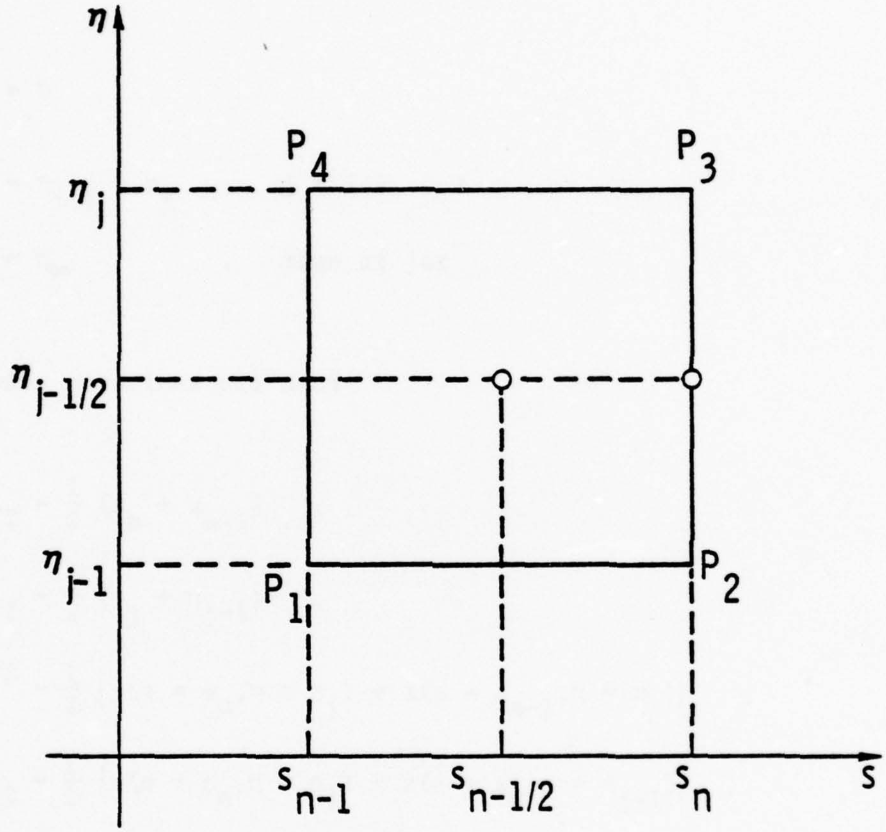


Fig. 2 Mesh configuration

$$\begin{cases} s_o = 0 & , \\ s_n = s_{n-1} + k_n & , \quad n = 1,2,\ldots,N \end{cases}$$

$$\begin{cases} \eta_o = 0 \\ \eta_j = \eta_{j-1} + h_j & , \quad j = 1,2,\ldots J \\ \eta_J = \eta_\infty & , \quad \text{edge of jet} \end{cases}$$

The following notations are used:

$$s_{n-1/2} = \frac{1}{2}(s_n + s_{n-1}) \quad ,$$

$$\eta_{j-1/2} = \frac{1}{2}(\eta_j + \eta_{j-1}) \quad ,$$

$$z_j^{n-1/2} = \frac{1}{2}\Big(z(s = s_n, \eta = \eta_j) + z(s = s_{n-1}, \eta = \eta_j)\Big) \quad ,$$

$$z_{j-1/2}^{n} = \frac{1}{2}\Big(z(s = s_n, \eta = \eta_j) + z(s = s_n, \eta = \eta_{j-1})\Big) \quad ,$$

$$\alpha_1 = 1 + \varepsilon_1 \quad ,$$

$$\alpha_2 = 1 + \varepsilon_2 \quad ,$$

$$P_1 = \left(\frac{s}{h_2}\frac{\partial h_2}{\partial s}\right)_{j-1/2}^{n-1/2} \quad ,$$

$$P_6 = \frac{s_{n-1/2}}{k_n}$$

We now derive the difference equations approximating system (3.1). From Fig. 2, consider box $P_1P_2P_3P_4$. Equations (3.1a-c) are approximated by centering about $(s_n, \eta_{j-1/2})$ of segment P_2P_3 (s_n is the streamwise station at which the solution vector (f,u,v,w,t,p) is to be computed):

$$\frac{f_j^n - f_{j-1}^n}{h_j} = u_{j-1/2}^n \tag{3.3a}$$

$$\frac{u_j^n - u_{j-1}^n}{h_j} = v_{j-1/2}^n \tag{3.3b}$$

$$\frac{w_j^n - w_{j-1}^n}{h_j} = t_{j-1/2}^n \tag{3.3c}$$

Next, Eqs. (3.1d-f) are approximated by centering about $(s_{n-1/2}, \eta_{j-1/2})$, the middle of the box $P_1P_2P_3P_4$:

$$\frac{(\alpha_1 v)_j^n - (\alpha_1 v)_{j-1}^n}{h_j} = -(1+P_1)(fv)_{j-1/2}^n - 2(u^2)_{j-1/2}^n + 4P_6\left(f_{j-1/2}^{n-1}v_{j-1/2}^n - f_{j-1/2}^n v_{j-1/2}^{n-1}\right)$$
$$+ 4P_6\left((u^2)_{j-1/2}^n - f_{j-1/2}^n v_{j-1/2}^n\right) + g_1^{n-1} \tag{3.3d}$$

$$\frac{(\alpha_2 t)_j^n - (\alpha_2 t)_{j-1}^n}{h_j} = -(1+P_1)(ft)_{j-1/2}^n - 4(K_1)_{j-1/2}^{n-1/2}(uw)_{j-1/2}^n - \sqrt{s}_{n-1/2}(K_2)_{j-1/2}^{n-1/2}(u^2)_{j-1/2}^n$$
$$+ 4P_6\left(w_{j-1/2}^n u_{j-1/2}^{n-1} + t_{j-1/2}^n f_{j-1/2}^{n-1}\right) + g_2^{n-1}$$
$$+ 4P_6\left(u_{j-1/2}^n w_{j-1/2}^n - f_{j-1/2}^n t_{j-1/2}^n\right) \tag{3.3e}$$

$$\frac{\tilde{p}_j^n - \tilde{p}_{j-1}^n}{h_j} = (\kappa)_{j-1/2}^{n-1/2}(u^2)_{j-1/2}^n + g_3^{n-1} \qquad (3.3f)$$

where g_1^{n-1}, g_2^{n-1}, and g_3^{n-1} (the dependent variables in g_1, g_2, g_3 are evaluated only on the previous streamwise station s_{n-1}) are given by

$$g_1^{n-1} = -\frac{(\alpha_1 v)_j^{n-1} - (\alpha_1 v)_{j-1}^{n-1}}{h_j} - (1 + P_1)(fv)_{j-1/2}^{n-1} - 2(u^2)_{j-1/2}^{n-1} + 4P_6\left(-(u^2)_{j-1/2}^{n-1} + f_{j-1/2}^{n-1}v_{j-1/2}^{n-1}\right) \qquad (3.3d')$$

$$g_2^{n-1} = -\frac{(\alpha_2 t)_j^{n-1} - (\alpha_2 t)_{j-1}^{n-1}}{h_j} - (1 + P_1)(ft)_{j-1/2}^{n-1} - 4(K_1)_{j-1/2}^{n-1/2}(uw)_{j-1/2}^{n-1}$$
$$- \sqrt{s_{n-1/2}}(K_2)_{j-1/2}^{n-1/2}(u^2)_{j-1/2}^{n-1} + 4P_6\left(-w_{j-1/2}^{n-1}(u_{j-1/2}^n + u_{j-1/2}^{n-1}) + t_{j-1/2}^{n-1}(f_{j-1/2}^{n-1} - f_{j-1/2}^n)\right)$$
$$(3.3e')$$

$$g_3^{n-1} = -\frac{(\tilde{p})_j^{n-1} - \tilde{p}_{j-1}^{n-1}}{h_j} + (\kappa)_{j-1/2}^{n-1/2}(u^2)_{j-1/2}^{n-1} \quad .$$

Equations (3.3a-3.3f) together with (3.3d'-f') are to be applied to all η-points, $j = 1,2,\ldots,J$.

The boundary conditions to be applied at $s = s_n$ are:

$$\text{wall} = \begin{cases} f_o^n = u_o^n = 0 & (3.4a) \\ w_0^n = 0 & (3.4b) \end{cases}$$

$$
\text{jet edge} = \begin{cases} f_J^n = F_{n_1}(s_n) & (3.4c) \\ W_J^n = F_{n_2}(s_n) & (3.4d) \\ P_J^n = F_n(s_n) \end{cases}
$$

3.2 Solution of the Difference Equations

Assuming solution is known at $s = s_{n-1}$, i.e., $(f_j^{n-1}, u_j^{n-1}, v_j^{n-1}, w_j^{n-1}, t_j^{n-1}, \tilde{p}_j^{n-1})$ for $0 \leq j \leq J$, we now want to evaluate the solution at $s = s_n$. We apply Eqs. (3.3) for $j = 1,\ldots,J$. Together with the boundary conditions (3.4), this yields 6*(J+1) equations for the 6*(J+1) unknowns $f_j^n, u_j^n, v_j^n, w_j^n, t_j^n, p_j^n)$, $j = 0,1,2,\ldots,J$.

For turbulent wall jets, the streamwise and spanwise momentum equations are coupled through the eddy viscosity of Eqs. (2.7). To handle this computationally, the streamwise momentum equation is solved using the $\partial w/\partial y$ associated with the previous s step as indicated in Section 2.6. This effectively decouples the streamwise momentum equation from the spanwise momentum equation at any station $s = s_n$, reducing the computational time and storage requirement.

With the above approximation, the solution algorithm is then given by:

$$
\begin{aligned}
&\text{(i) Solve for } (f_j^n, u_j^n, v_j^n) \quad , \quad j = 0,1,2,\ldots,J \quad . \\
&\text{(ii) Use (i) to solve for } (w_j^n, t_j^n) \quad , \quad j = 0,1,2,\ldots,J \qquad (3.5) \\
&\text{(iii) Use (i) to compute } p_j^n \quad , \quad j = 0,1,2,\ldots,J \quad .
\end{aligned}
$$

To treat (i), we must solve a system of 3*(J+1) nonlinear equations. (The equations are $(3.3a,b,d)_{j=1}^{J}$ and (3.4a,c).) Then, assuming (f,u,v) is successfully computed at $s = s_n$, (ii) and (iii) involves only systems of linear equations. Thus, the major bulk of computational time is in (i).

We note that the difference equations (3.3) for the variables $(f_j^n, u_j^n, v_j^n, w_j^n, t_j^n, p_j^n)$ for $j = 0,1,2,\ldots,J$ and $s = s_n$ can be viewed as the solution to two-point boundary value problems of systems of linear or non-linear ordinary differential equations, with the independent variable being η. Thus (i) now can be viewed as solution to:

$$\frac{df}{d\eta} = u \tag{3.6a}$$

$$\frac{du}{d\eta} = v \tag{3.6b}$$

$$\frac{d(\alpha,v)}{d\eta} = -(1 + P_1)fv - 2u^2 + 4P_6(u^2 - fv) + g_1(\eta) \tag{3.6c}$$

with boundary conditions

$$f(0) = u(0) = 0 \tag{3.6d}$$

$$u(\infty) = \text{constant} \tag{3.6e}$$

we have deliberately suppressed the dependence of s in P_1, P_6, g_1, and the constant in (3.6e). However, they do change as we march downstream.

The theory of numerical solution to two-point boundary value problems for ordinary differential equations can be found in Refs. 19 and 20. We shall only outline the procedure here.

The nonlinear system of equations for the unknown $\underset{\sim}{U} \equiv (f_j^n, u_j^n, v_j^n)_{j=o}^J$ are to be solved by Newton's method. Specifically, we define $\Phi(\underset{\sim}{U})$, (suppressing the s-dependence in (f,u,v) again),

$$0 = \Phi \equiv \begin{pmatrix} f_0 \\ u_0 \\ \dfrac{f_1 - f_0}{h_1} - \dfrac{u_0 + u_1}{2} \\ \dfrac{u_1 - u_0}{h_1} - \dfrac{v_0 + v_1}{2} \\ \dfrac{\alpha_1 v_1 - \alpha_o v_o}{h_1} + (1+P_1)\left(\dfrac{f_0+f_1}{2}\right)\left(\dfrac{v_0+v_1}{2}\right) + 2\left(\dfrac{u_0+u_1}{2}\right)^2 - 4P_6\left\{\left(\dfrac{u_0+u_1}{2}\right)^2 - \left(\dfrac{f_0+f_1}{2}\right)\left(\dfrac{v_0+v_1}{2}\right)\right\} - g_1(\eta_{1/2}) \\ \vdots \\ \dfrac{f_J - f_{J-1}}{h_J} - \dfrac{u_{J-1}+u_J}{2} \\ \dfrac{u_J - u_{J-1}}{h_J} - \dfrac{v_{J-1}+v_J}{2} \\ \dfrac{\alpha_J v_J - \alpha_{J-1} v_{J-1}}{h_J} + (1+P_1)\left(\dfrac{f_{J-1}+f_J}{2}\right)\left(\dfrac{u_{J-1}+u_J}{2}\right) + 2\left(\dfrac{u_{J-1}+u_J}{2}\right)^2 - 4P_6\left\{\left(\dfrac{u_{J-1}+u_J}{2}\right)^2 - \left(\dfrac{f_{J-1}+f_J}{2}\right)\left(\dfrac{v_{J-1}+u_J}{2}\right)\right\} \\ - g_1(\eta_{J-1/2}) \\ u_J - \text{constant} \end{pmatrix} \tag{3.7}$$

Let the initial iterate $\underset{\sim}{U}^{(o)}$ be the solution at the previous streamwise station s_{n-1}. Then, Newton's method* gives

$$\frac{\partial \Phi}{\partial \underset{\sim}{U}}^{(\nu-1)} \delta \underset{\sim}{U}^{(\nu-1)} = -\underset{\sim}{\Phi}\left(\underset{\sim}{U}^{(\nu-1)}\right) \quad , \quad \nu \geq 1 \tag{3.8a}$$

$$\underset{\sim}{U}^{(\nu)} = \underset{\sim}{U}^{(\nu-1)} + \delta \underset{\sim}{U}^{(\nu-1)} \quad , \quad \nu \geq 1 \tag{3.8b}$$

Method is said to have converged at the Kth iteration when

$$\| \delta \underset{\sim}{U}^{(K-1)} \| \leq \text{prescribed error tolerance} \tag{3.9}$$

The Jacobian matrix $\partial \underset{\sim}{\Phi} / \partial \underset{\sim}{U}$ in (3.8a) has a very nice structure, a consequence of the centered-Euler method in approximating (3.6)

$$\frac{\partial \Phi}{\partial \underset{\sim}{U}} \equiv \begin{bmatrix} \begin{pmatrix} 100 \\ 010 \end{pmatrix} & & & & \\ \mathcal{L}_1 & R_1 & & 0 & \\ & \mathcal{L}_2 & R_2 & & \\ & & \ddots & \ddots & \\ 0 & & & \mathcal{L}_J & R_J \\ & & & 0 & (010) \end{bmatrix} \tag{3.10}$$

*When α_K depends on U_ℓ , $\ell \neq K-1,K$, as most all eddy viscosity models do, then we do not have Newton's method strictly speaking, because we avoid terms $[(\partial\alpha_K/\partial u)\delta u]v$ and $[(\partial\alpha_K/\partial v)\delta v]v$.

where $\mathcal{L}_K$, R_K are (3×3) matrices, K = 1,2,...,J, are given by:

$$\mathcal{L}_K \equiv \begin{bmatrix} -\frac{1}{h_K} & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{h_K} & -\frac{1}{2} \\ -\beta_1^K & -\beta_2^K & -\frac{\alpha_{K-1}}{h_K} - \beta_3^K \end{bmatrix} , \tag{3.11a}$$

$$R_K \equiv \begin{bmatrix} +\frac{1}{h_K} & \frac{1}{2} & 0 \\ 0 & +\frac{1}{h_K} & \frac{1}{2} \\ -\beta_1^K & -\beta_2^K & +\frac{\alpha_{K-1}}{h_K} - \beta_3^K \end{bmatrix} , \tag{3.11b}$$

with

$$\begin{aligned} \beta_1^K &= -\frac{1}{2}\left\{(1+P_1)v_{K-1/2} + 4P_6 \frac{1}{2} v_{K-1/2}\right\} , \\ \beta_2^K &= \frac{1}{2}\left\{-2u_{K-1/2} + 4P_6 u_{K-1/2}\right\} , \\ \beta_3^K &= -\frac{1}{2}\left\{(1+P_1)f_{K-1/2} + 4P_6 f_{K-1/2}\right\} \end{aligned} \tag{3.12}$$

The first two rows in (3.10) are contributions from boundary conditions at the wall, with the last row from the jet edge. $\partial \underline{\Phi}/\partial \underline{U}$ can be further partitioned into a block tridiagonal matrix $[B_i A_i C_i]$, where

$$B_i = \begin{bmatrix} x & x & x \\ x & x & x \\ 0 & 0 & 0 \end{bmatrix}, \qquad i \geq 1 \tag{3.13a}$$

$$A_i = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}, \qquad i \geq 1 \tag{3.13b}$$

$$C_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ x & x & x \end{bmatrix}, \qquad i \geq 1 \tag{3.13c}$$

Before we go to the next section to describe an algorithm for solving such a matrix system, we want to comment on the solution procedure for (ii) and (iii) of (3.5). As remarked earlier, since (f,u,v) are now known at $s = s_n$, the equations for $w_j^n, t_j^n)$, $j = 0,1,2,\ldots,J$ can be viewed as the difference approximation to the <u>linear</u> two-point boundary value problem in <u>one</u> independent variable η of the form

$$\frac{dZ}{d\eta} = A(\eta)\underset{\sim}{Z} + \underset{\sim}{g}(\eta) \tag{3.14a}$$

$$B_0\underset{\sim}{Z}(0) = 0 \tag{3.14b}$$

$$B_1\underset{\sim}{Z}(\eta_\infty) = \text{constant} \tag{3.14c}$$

with $\underset{\sim}{Z} \equiv (w,t)^T$, A has coefficients (f,u,v) and $\underset{\sim}{g} = (0,g_2)$, the system of <u>linear</u> equations can again be partitioned into the form

$$\mathbb{A}\,\underset{\sim}{Z}_h = \underset{\sim}{b}_h$$

where $\mathbb{A}$ is a block tridiagonal matrix, $\underset{\sim}{Z}_h$ and $\underset{\sim}{b}_h$ are given by

$$\underset{\sim}{Z}_h \equiv \begin{pmatrix} w_0 \\ t_0 \\ w_1 \\ t_1 \\ w_2 \\ t_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ w_J \\ t_J \end{pmatrix} , \quad \underset{\sim}{b}_h \equiv \begin{pmatrix} 0 \\ 0 \\ g_2(\eta_{1/2}) \\ 0 \\ g_2(\eta_{j-1/2}) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ g_2(\eta_{J-1/2}) \\ 0 \end{pmatrix} .$$

The computation of (iii) is simply the centered-Euler integration of

$$\tilde{p}(\eta_K) = \tilde{p}(\infty) - \int_{\eta_K}^{\infty} \kappa u^2(\tau)d\tau \quad .$$

3.3 Block Tridiagonal Solver

In this section we describe the solution to

$$\mathbb{A}\,\underset{\sim}{x} = \underset{\sim}{b} \tag{3.15}$$

where

$$\mathbb{A} \equiv \begin{bmatrix} A_1 & C_1 & & & \\ B_2 & A_2 & C_2 & 0 & \\ & \ddots & \ddots & \ddots & \\ & & B_{J-1} & A_{J-1} & C_{J-1} \\ 0 & & & B_J & A_J \end{bmatrix} \equiv [B_i, A_i, C_i] \tag{3.16}$$

$$x \equiv \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ x_{3,1} \\ x_{1,2} \\ x_{2,2} \\ x_{3,2} \\ \vdots \\ x_{1,J} \\ x_{2,J} \\ x_{3,J} \end{bmatrix}, \quad b \equiv \begin{bmatrix} b_{1,1} \\ b_{2,1} \\ b_{3,1} \\ b_{1,2} \\ b_{2,2} \\ b_{3,2} \\ \vdots \\ b_{1,J} \\ b_{2,J} \\ \vdots \\ b_{n,J} \end{bmatrix} \tag{3.17}$$

and A_K, B_K, C_K are matrices of order n, K = 1,...,J.

$\mathbb{A}$ can be decomposed into the form

$$\mathbb{A} = \mathcal{L}\mathcal{U} \equiv [\beta_i \quad I \quad 0] * [0 \quad \alpha_i \quad C_i] \tag{3.18}$$

where

$$\alpha_1 = A_1 \quad , \tag{3.19a}$$

$$\begin{cases} \beta_i \alpha_{i-1} = B_i & , \quad i = 2,3,\ldots,J \\ \alpha_i = A_i - \beta_i C_{i-1} & , \quad i = 2,3,\ldots,J \end{cases} \tag{3.19b}$$

Here matrices α_i in turn are decomposed into the form:

$$\alpha_i = p_i \ell_i u_i q_i \tag{3.20}$$

where p_i, q_i are permutation matrices for row-and-column pivoting. ℓ_i and u_i are lower and upper triangular matrices. It is important to have an accurate LU-factorization of α_i because they are used in solving both B_{i+1} and x_i. Here we use a mixed pivoting strategy. During the Kth stage of Gaussian elimination, the pivot $a_{kk}^{(K)}$ is chosen to satisfy

$$|a_{kk}^{(K)}| \geq |a_{k,\ell}^{(K)}| \quad , \quad |a_{\ell,k}^{(K)}| \qquad \ell > k \quad . \tag{3.21}$$

This mixed pivoting is much better than the partial column pivoting or partial row pivoting. If ε_{im} is the round-off error using the mixed pivoting strategy, and if ε_{ip} is the round-off error using either partial column or partial row pivoting strategy, then it can be easily shown that

$$\|\varepsilon_{ip}\| \geq \|\varepsilon_{im}\| \tag{3.22}$$

The solution of β_i from Eq. (3.19b) can be easily carried out. Assuming there are p rows of B_i with at least one non-zero element on that row, then the solution of β_i corresponds to inverting the following

$$\left.\begin{aligned} \tilde{u}_{i-1}^T \underset{\sim}{y}_K &= \underset{\sim}{B}_{i,K}^T \\ \tilde{\ell}_{i-1}^T \underset{\sim}{\beta}_K^T &= \underset{\sim}{y}_K \end{aligned}\right\} \qquad K = 1,2,\ldots,p \tag{3.23}$$

where $\tilde{\ell}_{i-1}$, $\tilde{u}_{i-1}$ are lower and upper triangular matrices of size n. We further note the zero structure (as in (3.13a)) is preserved under such a decomposition scheme. This avoids unnecessary storage space requirements.

Now assume (3.19) has been performed, then to solve x, we merely have to solve

$$\mathcal{L}\underset{\sim}{z} = \underset{\sim}{b} \tag{3.24}$$

$$U\underset{\sim}{x} = \underset{\sim}{z} \tag{3.25}$$

where $z \equiv (\underset{\sim}{z}_1,\underset{\sim}{z}_2,\ldots,\underset{\sim}{z}_J)^T$, $\underset{\sim}{z}_\ell = (z_1,z_2,\ldots,z_n)^T$ can be obtained:

$$\underset{\sim}{z}_1 = \underset{\sim}{b}_1 \tag{3.26a}$$

$$\underset{\sim}{z}_K = \underset{\sim}{b}_K - \beta_K \underset{\sim}{z}_{K-1} \qquad K = 2,3,\ldots,J \tag{3.26b}$$

and (3.25) will give the solution vector $\underset{\sim}{x}$

$$\alpha_J x_J = z_J \tag{3.27a}$$

$$\alpha_{\ell-1} x_{\ell-1} = z_{\ell-1} - c_{\ell-1} x_\ell \qquad \ell = J,J-1,\ldots,2 \quad . \tag{3.27b}$$

3.4 Starting Procedure

The starting procedure is to employ a suitable discretization of the initial conditions obtained by the method described in Section 2.7. Glauert's similarity solution derived in Ref. 1 was implemented using the s = 0 specialization of Eqs. (3.3) and (3.4) for the results given in this report. However, with the compatibility restrictions given in Section 2.7, more general initial conditions could be accommodated including those derived from experimental data.

4.0 PARAMETRIC STUDIES

In this section, results will be indicated typifying calculations using the computational model. Because of its interest on the XFV-12A augmenter, a logarithmic wing spiral contour, shown schematically in Fig. 3, will be discussed. In contrast to previously published solutions exemplified by Ref. 3, this discussion will deal with non-similar flows due to the nature of the assumed turbulence model. In Ref. 3, the logarithmic spiral shape with certain assumptions on the scaling of jet thickness with downstream distance gave rise to similitude and an analytic solution for the flow. The non-similar framework considered here makes such a solution unlikely and numerical methods must be used. In the notation of the figure, the equation of the spiral contour is

$$s = s_0 e^{\theta/K} \quad , \tag{4.1}$$

where s is the running arc length, θ is the local inclination of the surface, where K and s_0 are constants. For $K > 0$, a convex contour is obtained, and with $K < 0$, concavity is implied. Equation (4.1) can be represented parametrically as

$$\frac{X}{R_0} = \left\{ e^{\theta/K}\left[\sin\theta + K^{-1}\cos\theta\right] - K^{-1}\right\} \Big/ (1+K^{-2}) \quad , \tag{4.1a}$$

$$\frac{Y}{R_0} = \left\{ e^{\theta/K}\left[K^{-1}\sin\theta - \cos\theta\right] + 1\right\} \Big/ (1+K^{-2}) \quad , \tag{4.2b}$$

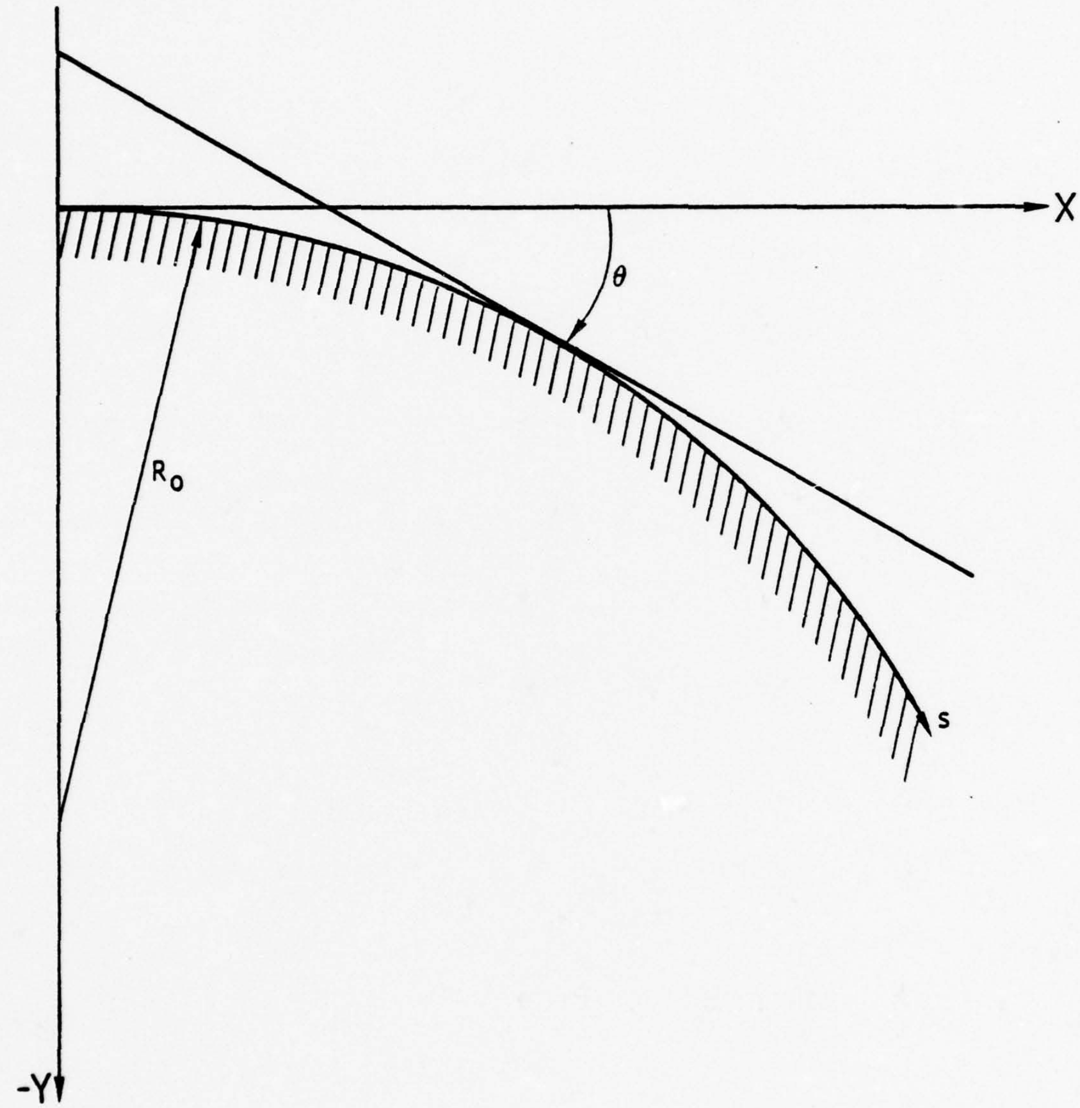


Fig. 3 Log spiral schematic

where R_0 is the initial radius of curvature and X and Y are Cartesian coordinates shown in Fig. 3. Wall shapes associated with K = 0.05 and K = 1/3 are depicted in Fig. 4.

In Fig. 5, the peak normalized streamwise velocity $f_{\eta_{max}}$ with the presence and absence of spanwise flow for a submerged wall jet ($f\eta(s,\infty) = 0$) is shown for K = 1/3. For computational convenience, the cross flow was generated by the forcing term K_2u^2 in (2.5d) with $w(0,\infty)$ assumed zero. This effect can be thought of as the influence of spanwise curvature on the w field and its interaction with the mainstream flow. It is rather obvious that for $K_2 = -5$ a small degradation occurs due to the turbulent coupling which is virtually imperceptible when the physical variable u_{MAX} is displayed. Another comparison shown for the effective mass entrainment function f(s,10) at the computational edge of the layer shows increased entrainment due to the cross flow.

Figure 6 indicates comparable small cross flow effects on the surface pressure distribution where the $K_2 = -5$ case is compared to w = 0 for the K = 1/3 log spiral. The lack of s^{-1} scaling is due to the non-similar nature of the assumed turbulence model.

In Figs. 7 and 8, the streamwise development of the u and w profiles is shown. Although both profiles resemble each other, the momentum in the cross flow increases, in contrast to the decay exhibited by u. This trend is also indicated in Fig. 9 for w_{MAX} and is due to the source-like manner in which the sidewash is produced.

To further assess the influence of turbulent coupling of the sidewash field on the mainstream flow, the effect of w on typical velocity profile is

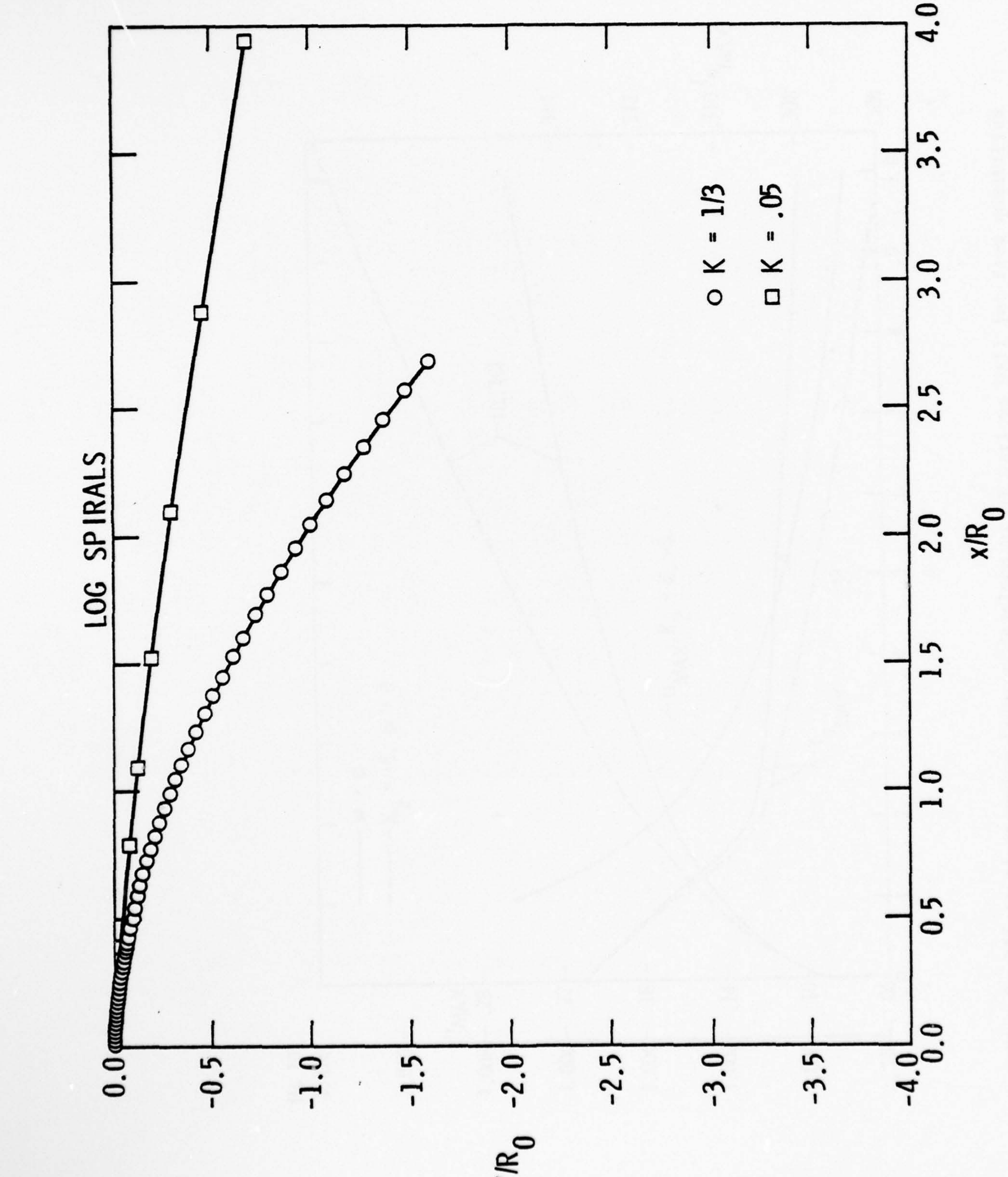


Fig. 4 Log spirals for K = .05 and K = 1/3

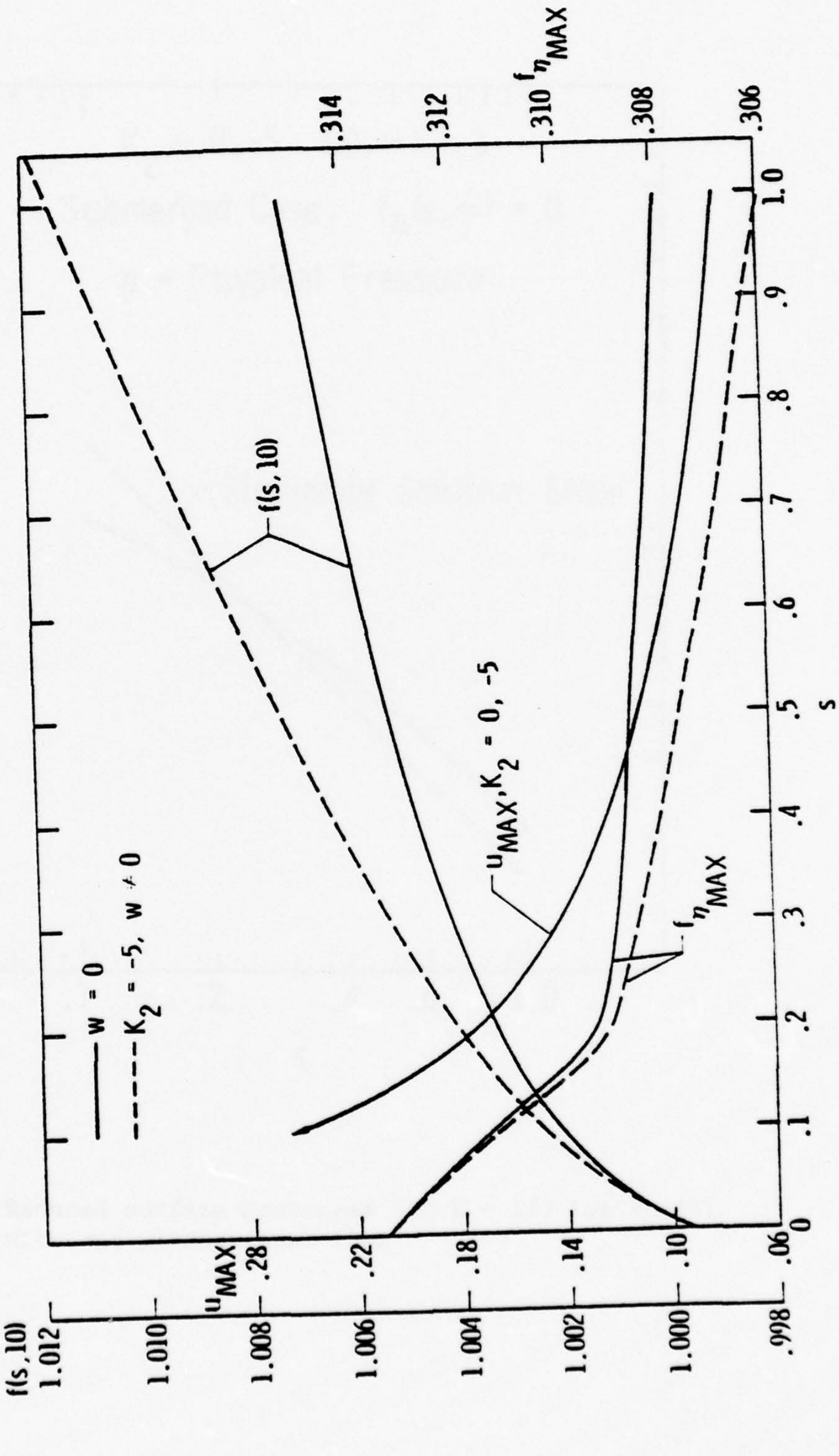


Fig. 5 Effect of spanwise flow on development of various wall jet flow quantities

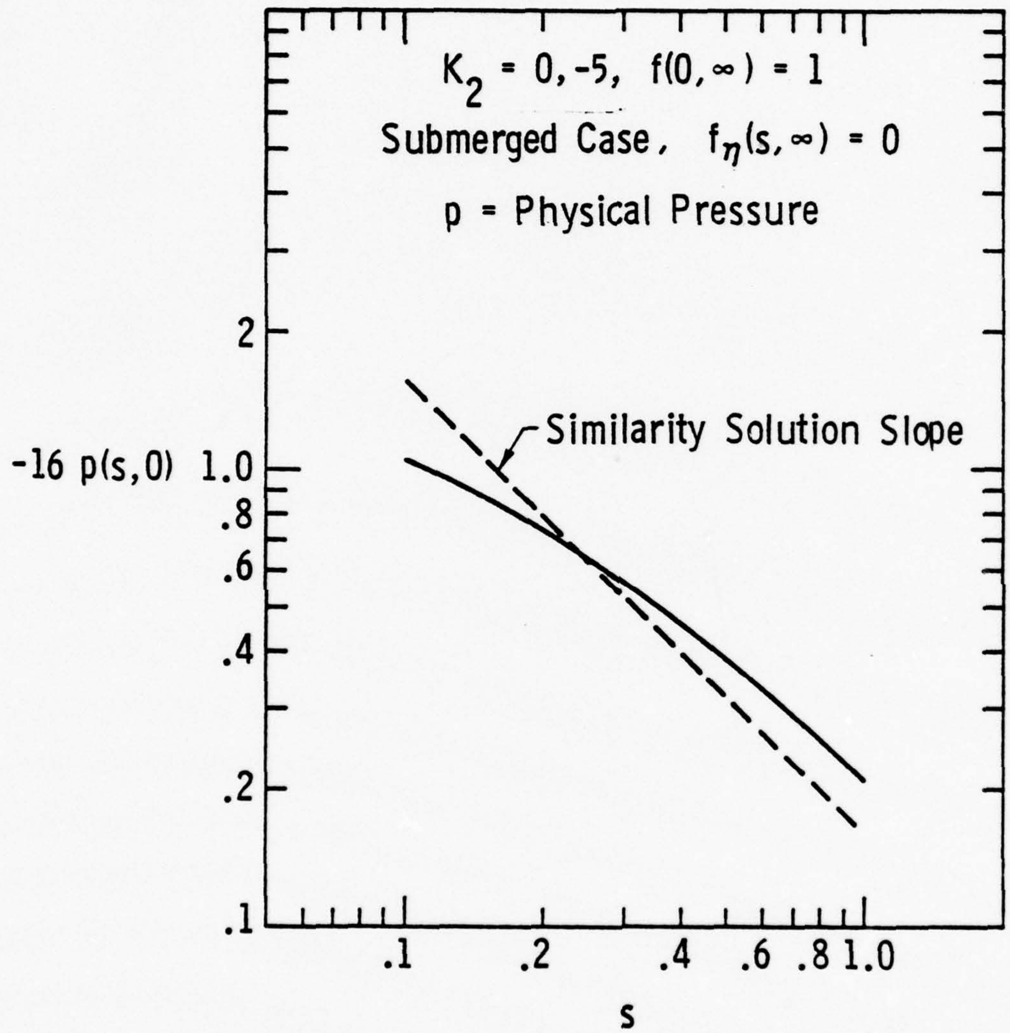


Fig. 6 Reduced surface pressures for K = 1/3 log spiral with and without span flow

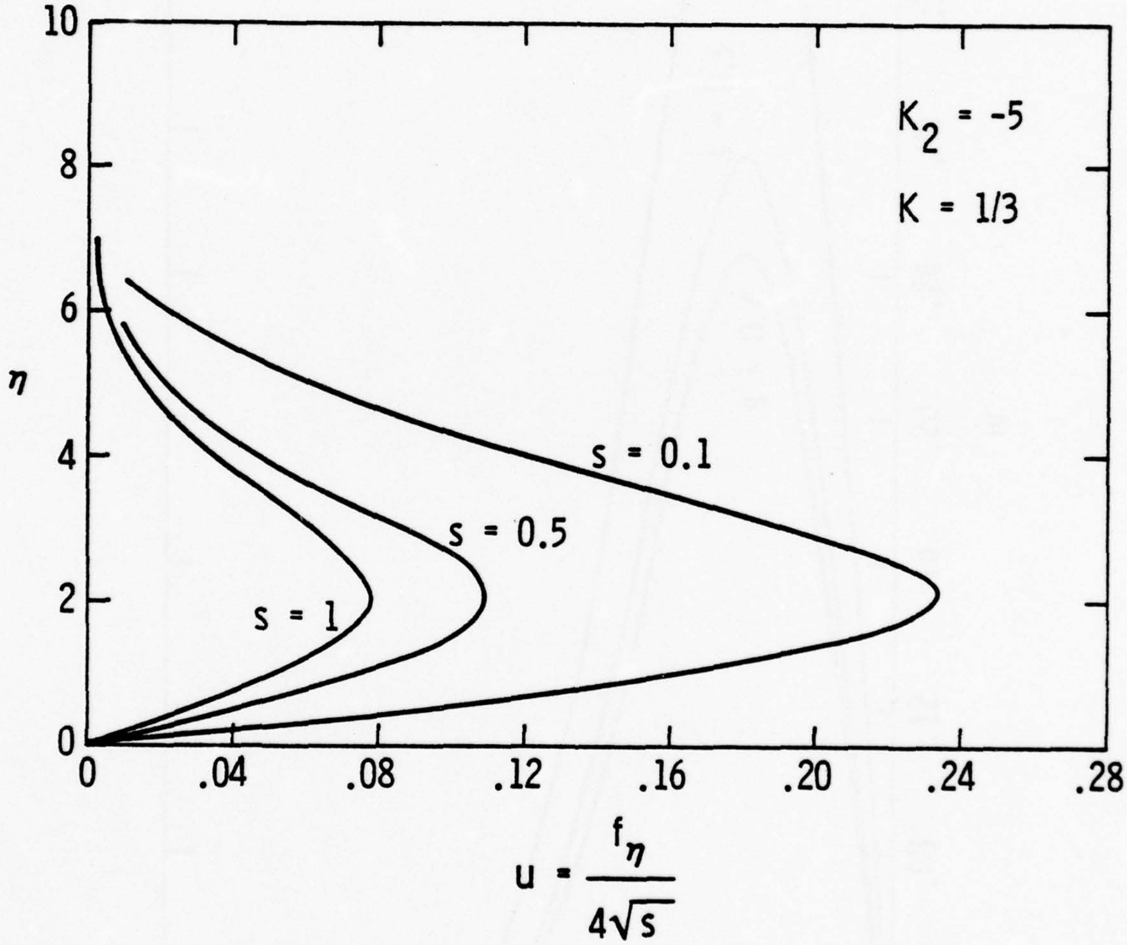


Fig. 7 Streamwise development of u profiles for log spiral-submerged wall jet with span flow

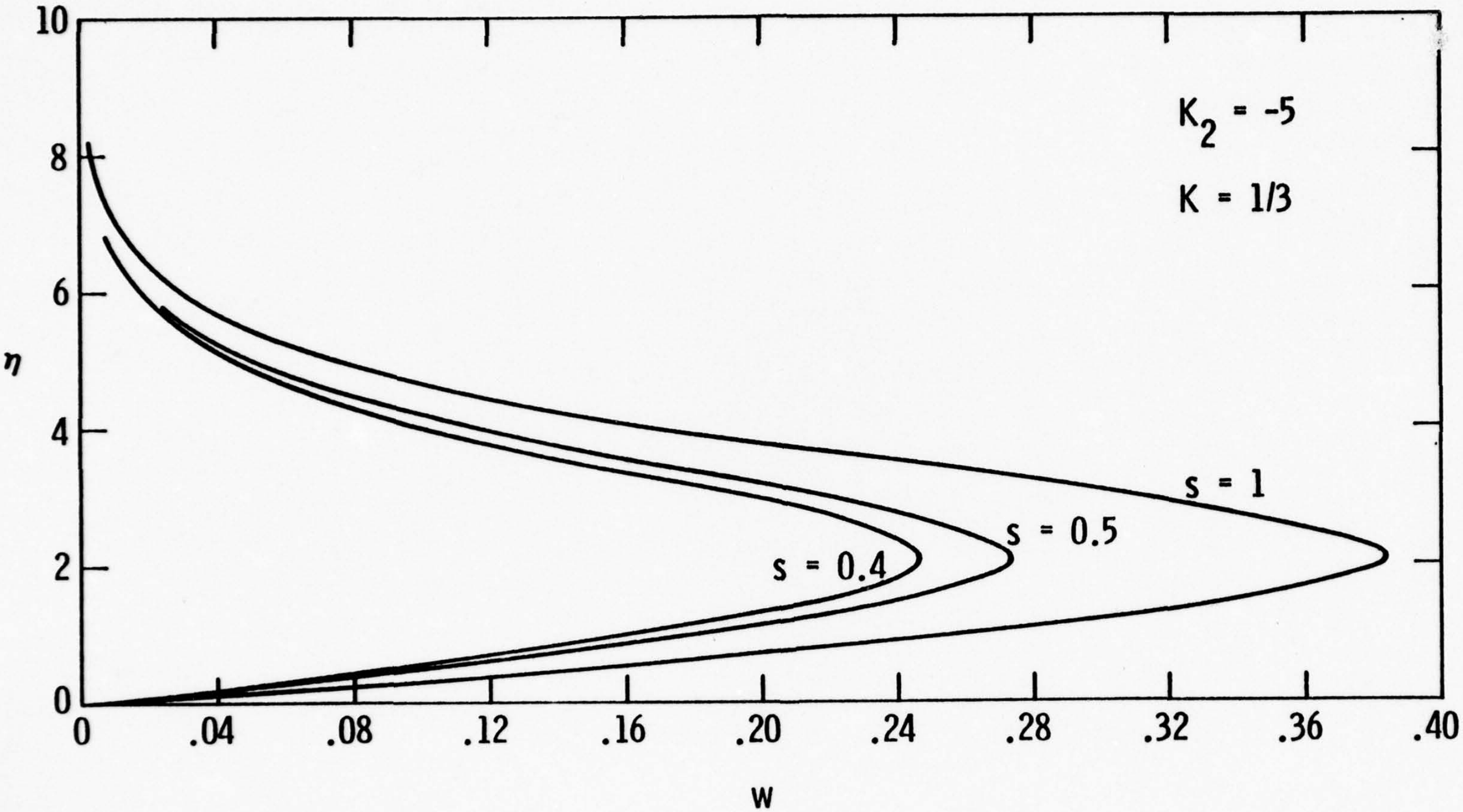


Fig. 8 Streamwise development of w profiles for log spiral-submerged wall jet

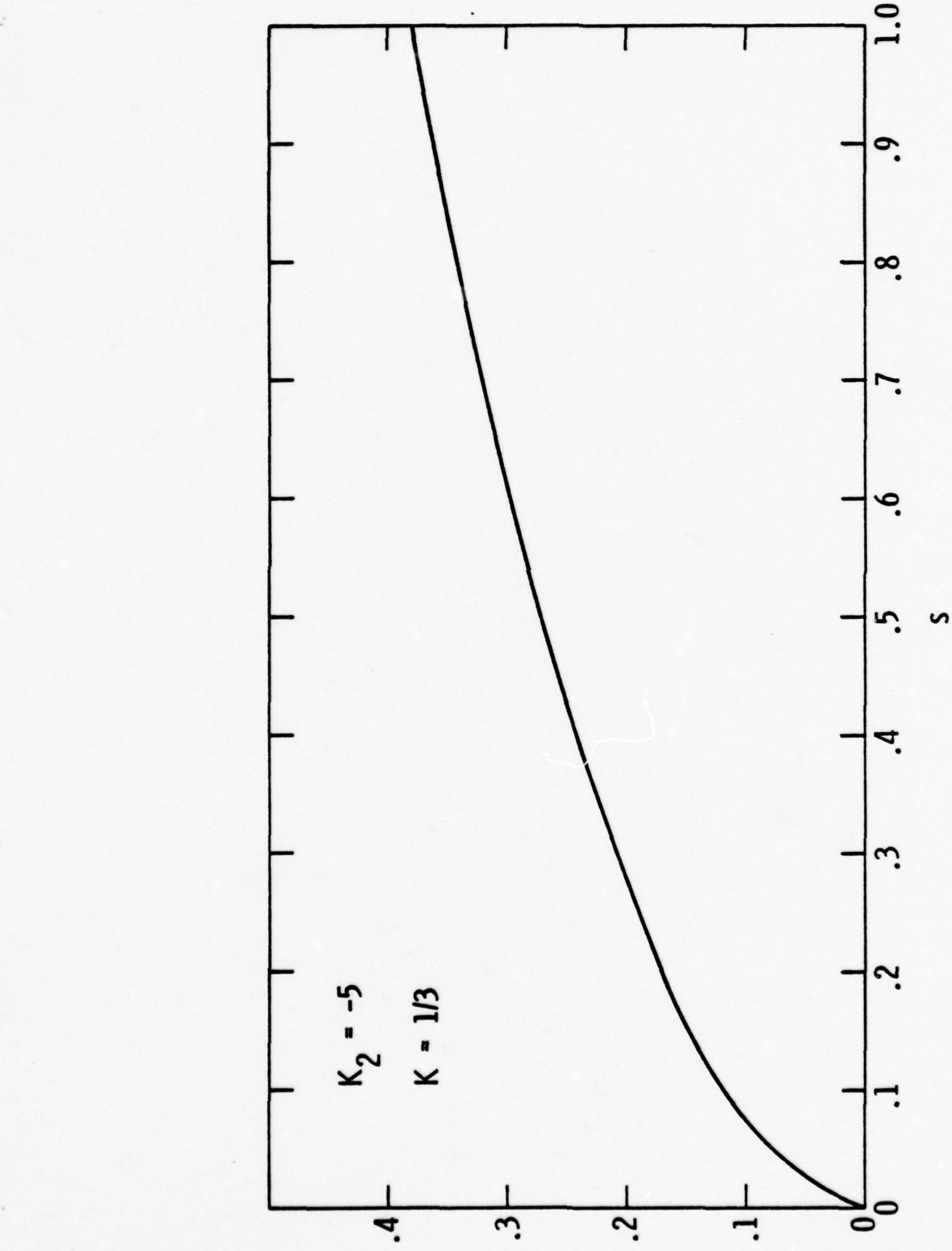


Fig. 9 Streamwise development of w_{MAX} for log spiral submerged wall jet

shown in Fig. 10, where the peak region is magnified to show the very small effect of the crossflow.

To illustrate the resemblence of velocity profile of coflowing wall jets and conventional boundary layers, a point made in Section 2.5, calculations were performed using the typical model of Eqs. (2.5) with $p = w = K_2 = f(0,0) = f_\eta(0,0) = 0$, and $u(x,\infty) = 1$. Results for the streamwise development of the reduced velocity profile and shear stress on a flat plate with $f(0,\infty) = 4$ are shown in Figs. 11 and 12. To indicate the potentialities of the existing code, the streamwise velocity profile development with downstream distance is shown in Fig. 13 for a logarithmic spiral contour with cross flow. Here, $K_2 = -10$, $u(x,\infty) = f(0,\infty) = 1$, and the external pressure gradient was neglected in the calculations. With this assumption, the qualitative downstream behavior resembles that of a flat plate.

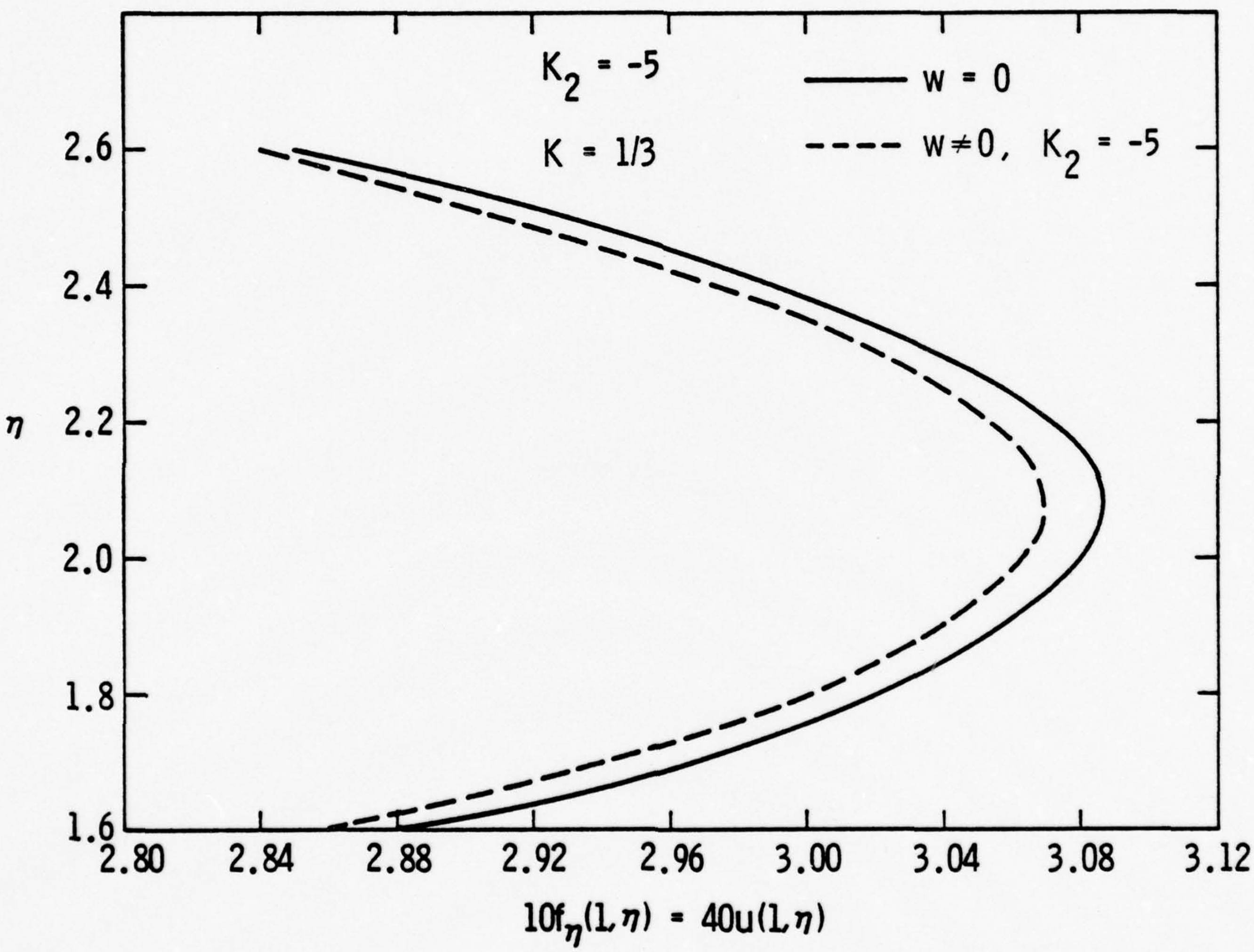


Fig. 10 Effect of spanwise flow on normalized u profile

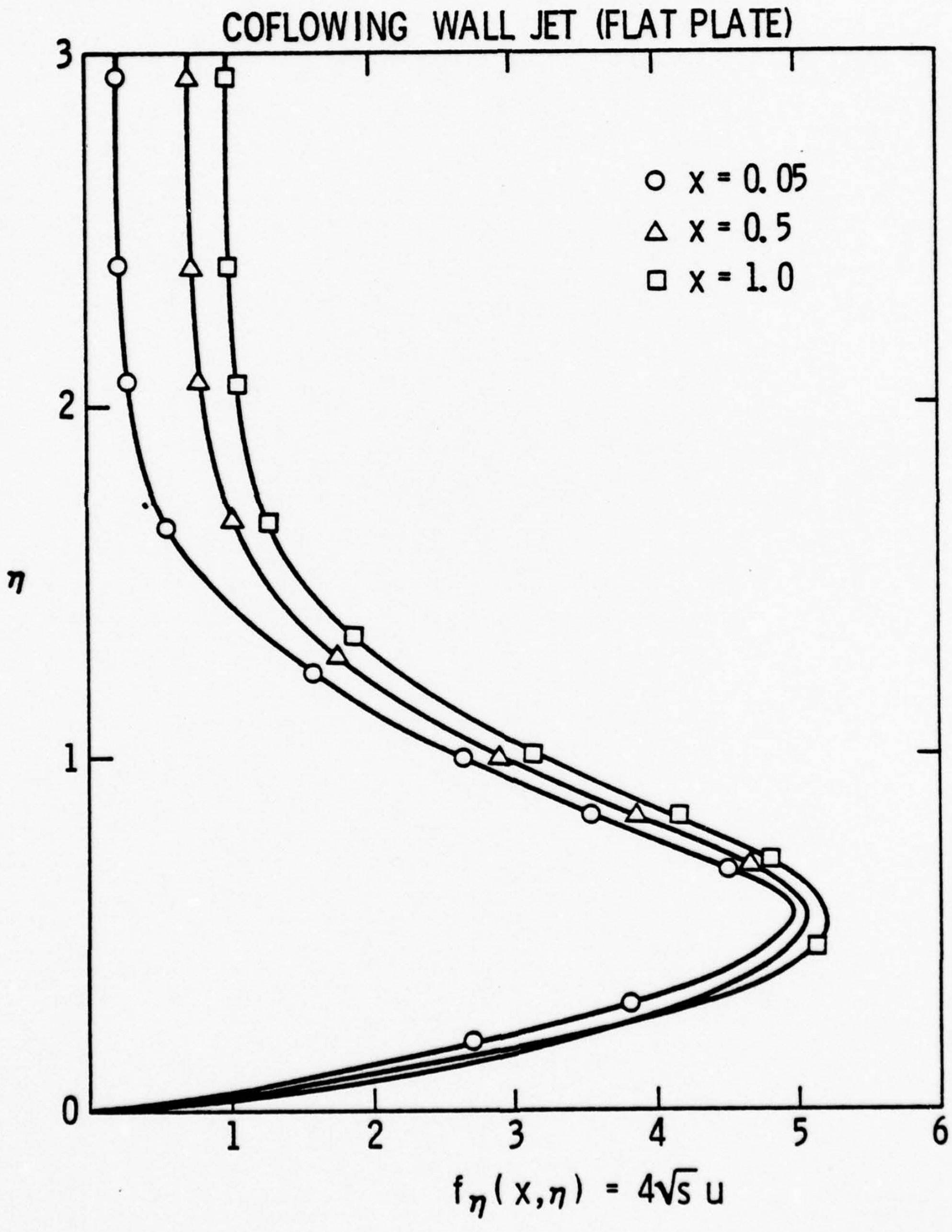


Fig. 11 Streamwise development of reduced velocity profile; $p = w = K_2 = f(0,0) = f_\eta(0,0) = 0,\ f(0,\infty) = 4$

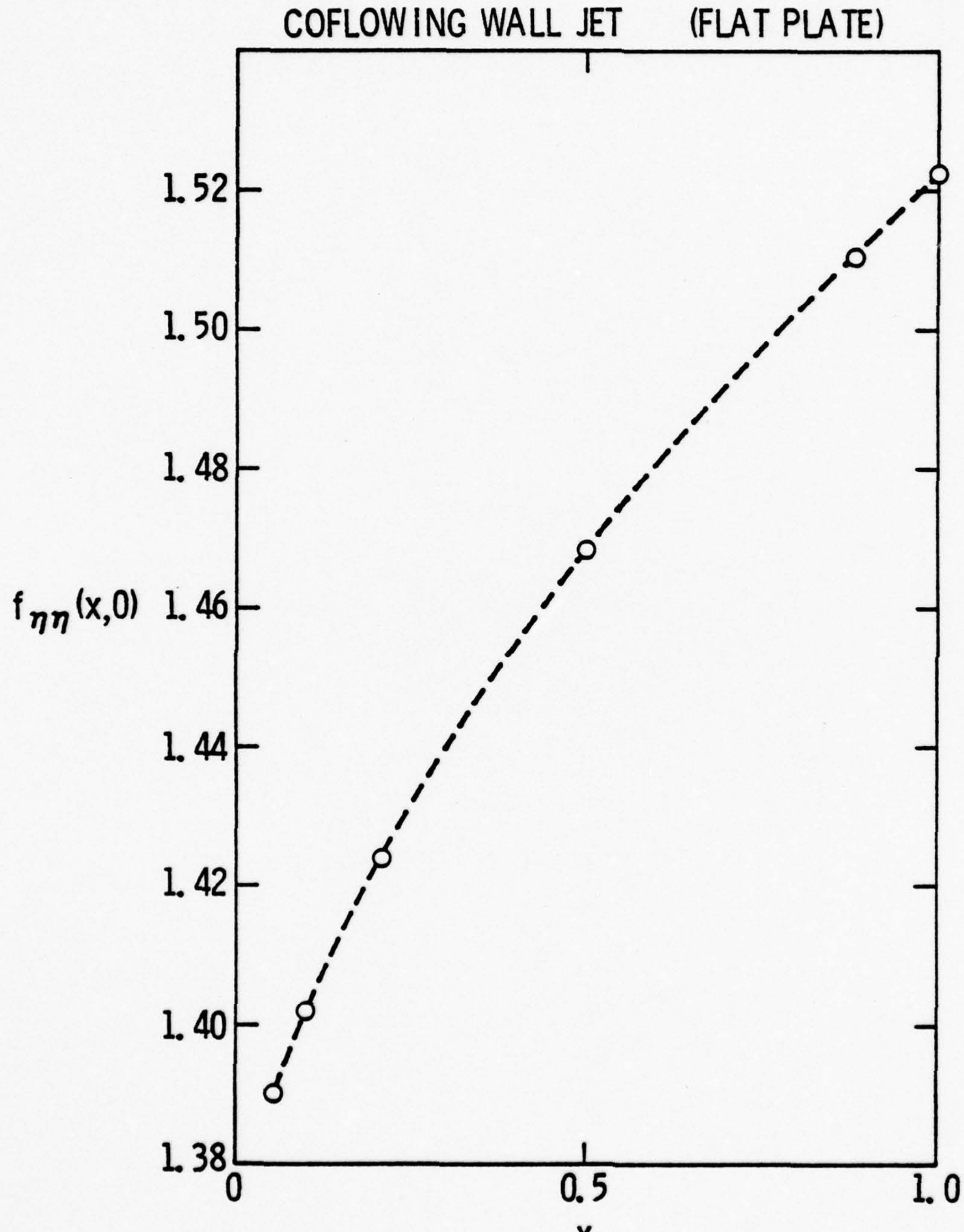


Fig. 12 Streamwise development of reduced wall shear stress; $p = w = K_2 = f(0,0) = f_\eta(0,0) = 0$, $f(0,\infty) = 4$

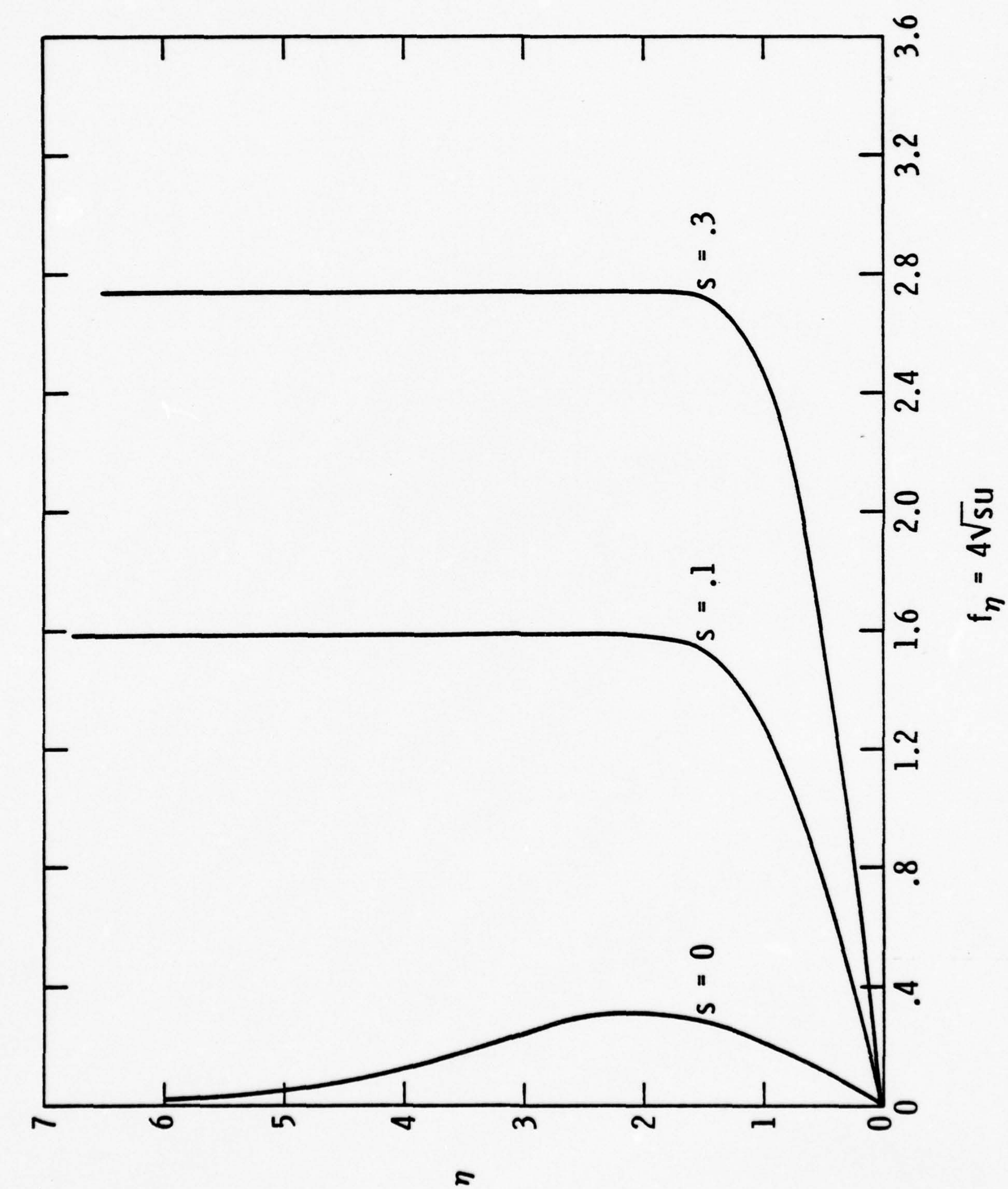


Fig. 13 Coflowing wall jet with cross flow $K_2 = -10$, $K = 1/3$ log spiral, $u(x,\infty) = f(0,\infty)' = 1$

3.0 CONCLUSIONS AND RECOMMENDATIONS

A computational model using Keller's box scheme has been developed to treat incompressible turbulent wall jets in a small cross flow approximation. The computer code can handle sidewash w injected as a source term in the spanwise momentum equation. The effect of the span flow on the streamwise flow is due to the eddy viscosity coupling between the u and w fields. For this type of spanwise flow generation, the coupling appears extremely weak, reducing the peak streamwise component and causing growth of w momentum in the downstream direction.

In subsequent effort, different modes of sidewash addition will be investigated, i.e., through the boundary and initial conditions. The results of this are indicative of flow conditions for wall jets on configurations with slight taper or sweep. For handling more realistic situations, the foregoing analysis will be extended to finite cross flow.

6.0 REFERENCES

[1] M. B. Glauert, "The Wall Jet," J. Fl. Mech. 1, 625-643 (1956).

[2] I. J. Wygnowski and F. H. Champagne, "The Laminar Wall Jet Over a Curved Surface," J. Fl. Mech. 31, Pt. 3, 459-465 (1968).

[3] J. A. Giles, A. P. Noyes, and R. A. Sawyer, "Turbulent Wall Jets on Logarithmic Spiral Surfaces," Aero Quarterly XVII, 202-215 (1965).

[4] R. J. Goldstein and D. J. Wilson, "Turbulent Wall Jets with Cylindrical Streamwise Surface Curvature," ASME Trans., J. Fluids Eng., 550-557 (1976).

[5] H. P. Irwin and P. A. Smith, "Prediction of the Effect of Streamline Curvature on Turbulence," Phys. of Fluids 18 (6), 624-630 (1975).

[6] R. M. C. So, "A Turbulence Velocity Scale for Curved Shear Flows," J. Fl. Mech. 70, Pt. 1, 37-57 (1975).

[7] I. P. Castro and P. Bradshaw, "The Turbulence Structure of a Highly Curved Mixing Layer," J. Fl. Mech. 73, Pt. 2, 265-304 (1976).

[8] V. Kruka and S. Eskinazi, "The Wall Jet in a Moving Stream," J. Fl. Mech. 20, Pt. 4, 555-579 (1964).

[9] N. V. Chandrasekhara Swamy and P. Bandyospadhyay, "Mean and Turbulence Characteristics of Three-Dimensional Wall Jets," J. Fl. Mech. 71, Pt. 3, 541-562 (1975).

[10] P. M. Sforza and G. Herbst, "A Study of Three-Dimensional Incompressible Turbulent Wall Jets," AIAA J. 8, 276-283 (1970).

[11] A. D. De Joode and S. V. Pantankar, "Prediction of Three-Dimensional Turbulent Mixing in an Ejector," AIAA Paper 77-706 (1977).

[12] F. A. Dvorak, "Calculation of Turbulent Boundary Layers and Wall Jets Over Curved Surfaces," AIAA J. 11, 517-524 (1973).

[13]S. Kaplun, Z. Angew Math u Mech., 111-135 (1954).

[14]P. A. Lagerstrom, "Laminar Flow," B, from Laminar Flows and Transition to Turbulence, Princeton Series on High Speed Aerodynamics and Jet Propulsion, Vol. IV (Princeton University Press, 1962).

[15]M. Van Dyke, "Higher Approximations in Boundary Layer Theory," Part 1, General Analysis, J. Fl. Mech. 14, 161-177 (1962).

[16]M. Van Dyke, Perturbation Methods in Fluid Mechanics (Academic Press, New York, 1964), p. 124.

[17]J. D. Cole, Perturbation Methods in Applied Mathematics (Blaisdell, Waltham, Mass, 1968).

[18]H. B. Keller, "A New Difference Scheme for Parabolic Problems," Numerical Solutions of Partial Differential Equations, Vol. II (1971).

[19]H. B. Keller, "Accurate Difference Methods for Nonlinear Two-Point Boundary Value Problems," SIAM J. Num. Anal. 11, 305-320 (1974).

[20]H. B. Keller, "Accurate Difference Methods for Linear Ordinary Differential Systems Subject to Linear Constraints," SIAM J. Num. Anal. 6, 8-30 (1969).

APPENDIX A: CODE LISTING AND SAMPLE INPUT AND OUTPUT

Sample Computer Output

STATION = 61. X = 1.00000E+00. X-STEP = 1.00000E-0
NEWTON - ITER = 5 ERROR = 5.06035E-07 AT(3, 3)   YIC = 2.70115E+01 U(3,1) 4.30071E+00 P = -5.77825E+02

	η	$f(s,\eta)$	$f_\eta(s,\eta)$	$f_{\eta\eta}$	$\bar{p}$	w	w_η
	Y	F	DF	DDF	P	W	DW
1	0.	9.6094661E-26	0.	4.3607074E+00	-5.7782494E+02	1.8110569E-16	2.3456261E+01
2	2.0000000E-01	1.7207544E-01	1.7207544E+00	7.8243808E+00	-5.7778980E+02	4.3030800E+00	1.4572594E+01
3	4.0000000E-01	6.4976453E-01	3.0559404E+00	5.5250633E+00	-5.7751408E+02	7.6412221E+00	1.3508762E+01
4	6.0000000E-01	1.3521503E+00	3.9651013E+00	3.5964052E+00	-5.7043470E+02	9.4201105E+00	6.4401227E+00
5	8.0000000E-01	2.2020217E+00	4.5300124E+00	2.0840601E+00	-5.7067754E+02	1.1336246E+01	5.2017115E+00
6	1.0000000E+00	3.1605350E+00	4.8425204E+00	9.7043008E-01	-5.7503507E+02	1.2090417E+01	2.3445197E+00
7	1.2000000E+00	4.1215254E+00	4.9673771E+00	2.7801324E-01	-5.7349444E+02	1.2404321E+01	6.5452305E-01
8	1.4000000E+00	5.1180039E+00	4.9974030E+00	2.2245613E-02	-5.7271732E+02	1.2471452E+01	1.0767449E-02
9	1.6000000E+00	6.1177248E+00	4.9990501E+00	2.2853505E-03	-5.7153234E+02	1.2470212E+01	-3.4193651E-02
10	1.8000000E+00	7.1177171E+00	5.0000104E+00	-6.7723038E-04	-5.7034665E+02	1.2402416E+01	-4.3740654E-02
11	2.0000000E+00	8.1177178E+00	4.9999496E+00	4.0459921E-04	-5.6916049E+02	1.2453726E+01	-4.3170077E-02
12	2.2000000E+00	9.1177173E+00	5.0000057E+00	-2.4337065E-04	-5.6797515E+02	1.2444757E+01	-4.6415205E-02
13	2.4000000E+00	1.0117718E+01	4.9999967E+00	1.5248155E-04	-5.6078934E+02	1.2433405E+01	-4.7205524E-02
14	2.6000000E+00	1.1117717E+01	5.0000020E+00	-9.8634254E-05	-5.6560304E+02	1.2425756E+01	-4.9260623E-02
15	2.8000000E+00	1.2117717E+01	4.9999967E+00	6.5347812E-05	-5.6441764E+02	1.2415796E+01	-5.0360053E-02
16	3.0000000E+00	1.3117717E+01	5.0000008E+00	-4.4040237E-05	-5.6323213E+02	1.2405595E+01	-5.1430484E-02
17	3.2000000E+00	1.4117717E+01	4.9999994E+00	3.0177011E-05	-5.6204638E+02	1.2395064E+01	-5.3070547E-02
18	3.4000000E+00	1.5117717E+01	5.0000004E+00	-2.0896849E-05	-5.6086063E+02	1.2384317E+01	-5.4340008E-02
19	3.6000000E+00	1.6117717E+01	4.9999997E+00	1.4613085E-05	-5.5967480E+02	1.2373328E+01	-5.5492553E-02
20	3.8000000E+00	1.7117717E+01	5.0000002E+00	-1.0306762E-05	-5.5048913E+02	1.2362113E+01	-5.6050277E-02
21	4.0000000E+00	1.8117717E+01	4.9999999E+00	7.3235342E-06	-5.5730337E+02	1.2350677E+01	-5.7670080E-02
22	4.2000000E+00	1.9117717E+01	5.0000001E+00	-5.2362462E-06	-5.5011702E+02	1.2339032E+01	-5.8752353E-02
23	4.4000000E+00	2.0117717E+01	4.9999999E+00	3.7689790E-06	-5.5493187E+02	1.2327184E+01	-5.9724787E-02
24	4.6000000E+00	2.1117717E+01	5.0000000E+00	-2.7261149E-06	-5.5374612E+02	1.2315141E+01	-6.0703550E-02
25	4.8000000E+00	2.2117717E+01	5.0000000E+00	1.9812560E-06	-5.5256037E+02	1.2302908E+01	-6.1524410E-02
26	5.0000000E+00	2.3117717E+01	5.0000000E+00	-1.4459830E-06	-5.5137461E+02	1.2290442E+01	-6.2532822E-02
27	5.2000000E+00	2.4117717E+01	5.0000000E+00	1.0592427E-06	-5.5018860E+02	1.2277649E+01	-6.3402762E-02
28	5.4000000E+00	2.5117717E+01	5.0000000E+00	-7.7859042E-07	-5.4900311E+02	1.2265133E+01	-6.4257020E-02
29	5.6000000E+00	2.6117717E+01	5.0000000E+00	5.7345028E-07	-5.4781736E+02	1.2252199E+01	-6.5081604E-02
30	5.8000000E+00	2.7117717E+01	5.0000000E+00	-4.2418392E-07	-5.4663161E+02	1.2239102E+01	-6.5584420E-02
31	6.0000000E+00	2.8117717E+01	5.0000000E+00	3.1418140E-07	-5.4544586E+02	1.2225045E+01	-6.6074327E-02
32	6.2000000E+00	2.9117717E+01	5.0000000E+00	-2.3309777E-07	-5.4426010E+02	1.2212433E+01	-6.7442731E-02
33	6.4000000E+00	3.0117717E+01	5.0000000E+00	1.7317450E-07	-5.4307435E+02	1.2198870E+01	-6.8191080E-02
34	6.6000000E+00	3.1117717E+01	5.0000000E+00	-1.2884996E-07	-5.4188860E+02	1.2185158E+01	-6.8424203E-02
35	6.8000000E+00	3.2117717E+01	5.0000000E+00	9.6015817E-08	-5.4070285E+02	1.2171302E+01	-6.9533997E-02
36	7.0000000E+00	3.3117717E+01	5.0000000E+00	-7.1613420E-08	-5.3951710E+02	1.2157304E+01	-7.0341557E-02
37	7.2000000E+00	3.4117717E+01	5.0000000E+00	5.3340493E-08	-5.3833134E+02	1.2143167E+01	-7.1027851E-02
38	7.4000000E+00	3.5117717E+01	5.0000000E+00	-3.9594257E-08	-5.3714559E+02	1.2128894E+01	-7.1700403E-02
39	7.6000000E+00	3.6117717E+01	5.0000000E+00	2.9332004E-08	-5.3595984E+02	1.2114448E+01	-7.2360309E-02
40	7.8000000E+00	3.7117717E+01	5.0000000E+00	-2.1840655E-08	-5.3477409E+02	1.2099951E+01	-7.3007540E-02
41	8.0000000E+00	3.8117717E+01	5.0000000E+00	1.6734030E-08	-5.3358834E+02	1.2085286E+01	-7.3642654E-02
42	8.2000000E+00	3.9117717E+01	5.0000000E+00	-1.3340705E-08	-5.3240259E+02	1.2070495E+01	-7.4260750E-02
43	8.4000000E+00	4.0117717E+01	5.0000000E+00	1.1166010E-08	-5.3121583E+02	1.2055591E+01	-7.4860013E-02
44	8.6000000E+00	4.1117717E+01	5.0000000E+00	-9.7047030E-09	-5.3003108E+02	1.2040544E+01	-7.5463170E-02
45	8.8000000E+00	4.2117717E+01	5.0000000E+00	8.5532740E-09	-5.2884533E+02	1.2025388E+01	-7.6076391E-02
46	9.0000000E+00	4.3117717E+01	5.0000000E+00	-7.5707692E-09	-5.2765950E+02	1.2010115E+01	-7.6060247E-02
47	9.2000000E+00	4.4117717E+01	5.0000000E+00	6.6587912E-09	-5.2647383E+02	1.1994725E+01	-7.7236478E-02
48	9.4000000E+00	4.5117717E+01	5.0000000E+00	-5.7679045E-09	-5.2528807E+02	1.1979221E+01	-7.7409470E-02
49	9.6000000E+00	4.6117717E+01	5.0000000E+00	4.8421600E-09	-5.2410232E+02	1.1963606E+01	-7.8350373E-02
50	9.8000000E+00	4.7117717E+01	5.0000000E+00	-3.9194517E-09	-5.2291557E+02	1.1947879E+01	-7.8907092E-02
51	1.0000000E+01	4.8117717E+01	5.0000000E+00	2.9725192E-09	-5.2173082E+02	1.1932043E+01	-7.9440527E-02
52	1.0800000E+01	5.2117717E+01	5.0000000E+00	-2.7336129E-09	-5.1694761E+02	1.1867620E+01	-8.1304024E-02
53	1.1600000E+01	5.6117717E+01	5.0000000E+00	2.5015249E-09	-5.1224480E+02	1.1801507E+01	-8.3522612E-02
54	1.2400000E+01	6.0117717E+01	5.0000000E+00	-2.2632271E-09	-5.0754180E+02	1.1733471E+01	-8.5460140E-02
55	1.3200000E+01	6.4117717E+01	5.0000000E+00	2.0833202E-09	-5.0275979E+02	1.1664699E+01	-8.7215421E-02
56	1.4000000E+01	6.8117717E+01	5.0000000E+00	-1.9034770E-09	-4.9801578E+02	1.1594417E+01	-8.8947585E-02
57	1.4800000E+01	7.2117717E+01	5.0000000E+00	1.7636049E-09	-4.9327277E+02	1.1522574E+01	-9.0717708E-02
58	1.5600000E+01	7.6117717E+01	5.0000000E+00	-1.6024010E-09	-4.8852977E+02	1.1447423E+01	-9.2257424E-02
59	1.6400000E+01	8.0117717E+01	5.0000000E+00	1.4782769E-09	-4.8378676E+02	1.1374999E+01	-9.3804179E-02
60	1.7200000E+01	8.4117717E+01	5.0000000E+00	-1.3679154E-09	-4.7904375E+02	1.1299341E+01	-9.5340750E-02
61	1.8000000E+01	8.8117717E+01	5.0000000E+00	1.2641699E-09	-4.7430074E+02	1.1222469E+01	-9.6764320E-02
62	2.2000000E+01	1.0811772E+02	5.0000000E+00	-1.2439337E-09	-4.5059571E+02	1.0820442E+01	-1.0345420E-01
63	2.6000000E+01	1.2811772E+02	5.0000000E+00	1.2250060E-09	-4.2087067E+02	1.0394466E+01	-1.0900349E-01
64	3.0000000E+01	1.4811772E+02	5.0000000E+00	-1.2086625E-09	-4.0315063E+02	9.9450175E+00	-1.1550077E-01
65	3.4000000E+01	1.6811772E+02	5.0000000E+00	1.1938926E-09	-3.7944080E+02	9.4758217E+00	-1.1949771E-01
66	3.8000000E+01	1.8811772E+02	5.0000000E+00	-1.1791982E-09	-3.5572550E+02	8.9872759E+00	-1.2477575E-01
67	4.2000000E+01	2.0811772E+02	5.0000000E+00	1.1646750E-09	-3.3201052E+02	8.4813504E+00	-1.2610049E-01
68	4.6000000E+01	2.2811772E+02	5.0000000E+00	-1.1518418E-09	-3.0829548E+02	7.9541173E+00	-1.3292957E-01
69	5.0000000E+01	2.4811772E+02	5.0000000E+00	1.1408914E-09	-2.8458045E+02	7.4213285E+00	-1.3540481E-01
70	5.4000000E+01	2.6811772E+02	5.0000000E+00	-1.1311453E-09	-2.6086541E+02	6.8693440E+00	-1.4002746E-01
71	5.8000000E+01	2.8811772E+02	5.0000000E+00	1.1227475E-09	-2.3715037E+02	6.3032406E+00	-1.4302432E-01
72	6.2000000E+01	3.0811772E+02	5.0000000E+00	-1.1157342E-09	-2.1343533E+02	5.7237504E+00	-1.4672070E-01
73	6.6000000E+01	3.2811772E+02	5.0000000E+00	1.1047382E-09	-1.8972030E+02	5.1317503E+00	-1.4917633E-01
74	7.0000000E+01	3.4811772E+02	5.0000000E+00	-1.1044817E-09	-1.6600526E+02	4.5282432E+00	-1.5265523E-01
75	7.4000000E+01	3.6811772E+02	5.0000000E+00	1.0944823E-09	-1.4229022E+02	3.9128176E+00	-1.5407260E-01
76	7.8000000E+01	3.8811772E+02	5.0000000E+00	-1.0941743E-09	-1.1857519E+02	3.2858445E+00	-1.5942341E-01
77	8.2000000E+01	4.0811772E+02	5.0000000E+00	1.0911327E-09	-9.4860149E+01	2.6447907E+00	-1.6060000E-01
78	8.6000000E+01	4.2811772E+02	5.0000000E+00	-1.0873584E-09	-7.1144112E+01	2.0023541E+00	-1.6121274E-01
79	9.0000000E+01	4.4811772E+02	5.0000000E+00	1.0840774E-09	-4.7430740E+01	1.3465415E+00	-1.6390172E-01
80	9.4000000E+01	4.6811772E+02	5.0000000E+00	-1.0411752E-09	-2.3715337E+01	6.7708036E-01	-1.6741214E-01
81	9.8000000E+01	4.8811772E+02	5.0000000E+00	1.0714924E-09	0.	0.	-1.7068059E-01

Sample Input Compilation

```
$PARMS
C1      =  0.33333333333333E+00,
C3      =  0.1E+01,
C7      = -0.1E+02,
HKS     =  0.1E-04,
FACX    =  0.12E+01,
```

```
      PROGRAM WALLJET(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C PROGRAM ARE TO BE RAN USING CDC 6600 CONTROL CARDS
C THIS IS THE MAIN PROGRAM FOR SOLVING THE THREE_DIMENSIONAL WALL_JET
C EQUATIONS WITH SMALL CROSS-FLOW APPROXIMATION.  UNDER THE WALL-JET
C APPROXIMATION, THE GOVERNING EQUATIONS OF MOTION BECOME PARABOLIC.
C A MARCHING TECNIQUE (IN THE STREAMWSIE DIRECTION) IS USED. THE BOX
C METHOD (REFERENCE 1) IS USED TO DISCRETISE THE NON-LINEAR WALL-JET
C EQUATIONS. THE DISCRETISED SYSTEM IS THEN SOLVED BY NEWTON#S METHOD.
C UNDER THE SMALL CROSS-FLOW ASSUMPTION, THE STREAMWISE AND SPANWISE
C MOMENTUM EQUATIONS BECOME UNCOUPLED. THIS ALLOWS SIMPLIFICATION IN
C THE SOLUTION ALGORITHM, AND REDUCTION IN BOTH STORAGE AND COMPUTAT-
C IONAL TIME. THE VARIABLES USED IN THE PROGRAM ARE (IF THE VARIABLES
C ARE INPUT PARAMETERS, THEY WILL BE DENOTED WITH ITS DEFAULT VALUE,
C IF ANY, ENCLOSED BY ** .... ** AT THE END OF THE DESCRIPTION OF SUCH
C VARIABLES, THEY ARE INPUT INTO PROGRAM BY NAMELIST ** INPUTS **.)
C
C        A    MAIN DIAGONAL BLOCKS OF THE BLOCK-TRIDIAGONAL MATRIX
C             (B A C) OBTAINED FROM LINEARISATION OF THE FINITE-DIFFERENCE
C             APPROXIMATION OF THE GOVERNING EQUATIONS
C      AJA    JACOBIAN MATRIX USED IN SETUP OF THE BLOCK TRIDIAGONAL
C             MATRIX
C        B    LOWER DIAGONAL BLOCKS, SEE DESCRIPTION OF VARIABLE A
C        C    UPPER DIAGONAL BLOCKS, SEE DESCRIPTION OF VARIABLE A
C       C1    REAL CONSTANT VARIABLE, K IN THE LOGARITHM SPIRAL
C             PARAMETRIC EQUATION, **INPUT, NO DEFAULT VALUE**
C       C3    REAL CONSTANT VARIABLE, INITAL MASS FLUX F(S=0,Y=INFINITY),
C             WHERE F IS THE GLAUERT SIMILARITY VARIABLE FOR THE STREAM-
C             WISE VELOCITY (REFERENCE 2), **INPUT, DEFAULT VALUE=1.**
C       C4    REAL CONSTANT VARIABLE. IN THE BOUNDARY CONDITION D/D(ETA)
C             (F) (S=ARBITRARY,ETA=INFINITY) =C4*(ARB. FN. OF S). C4 IS
C             NONZERO FOR COFLOWING WALL-JET, AND THE ARBITRARY FUNCTION
C             CAN BE CHANGED IN THE SUBROUTINE NAMED  BC BY THE USER,
C             **INPUT, NO DEFAULT VALUE**
C       C7    REAL CONSTANT VARIABLE, NONZERO FOR NON-ZERO CROSS-FLOW,
C             **INPUT, NO DEFAULT VALUE**
C    CASEW    LOGICAL VARIABLE, =.TRUE. IF THE SPANWISE MOMENTUM
C             EQUATION IS TO BE SOLVED, **INPUT, NO DEFAULT VALUE**
C   CUTOFF    REAL CONSTANT VARIABLE, USED IN CONVERGENCE TEST OF
C             NEWTON#S METHOD
C       DD    REAL VECTOR VARIABLE, THE DIFFERENCE BETWEEN TWO NEWTON
C             ITERATES
C   EPSERR    REAL CONSTANT VARIABLE, CONVERGENCE CRITERION OF NEWTON#S
C             METHOD
C        F    REAL VECTOR VARIABLE, RIGHT HAND SIDE OF THE BLOCK TRI-
C             DIAGONAL SYSTEM OF EQUATIONS
C     FACX    REAL CONSTANT VARIABLE, MULTIPLICATION FACTOR IN SETUP OF
```

```
C             INTERCHANGABLY), **INPUT, DEFAULT VALUE=1.2**
C       FF    REAL VECTOR VARIABLE, USED IN SETUP OF BLOCK TRIDIAGONAL
C             SYSTEM OF EQUATIONS
C       G     REAL VECTOR VARIABLE, USED IN SETUP OF BLOCK TRIDIAGONAL
C             SYSTEM OF EQUATIONS ( FIRST TWO COMPONENTS CONTAIN EDDY-
C             VISCOSITY INFORMATION, AND THE LAST TWO COMPONENTS CONTAIN
C             CONTRIBUTION OF THE PREVIOUS STREAMWISE STATION)
C       H     REAL VECTOR VARIABLE, VERTICAL MESH
C     HKS     REAL CONSTANT VARIABLE, INITIAL MESH-SIZE IN SETUP OF
C             STREAMWISE MESH,**INPUT, DEFAULT VALUE=1.E-5**
C     HKX     REAL VECTOR VARIABLE, STREAMWISE MESH
C    HMAX     REAL CONSTANT VARIABLE, MAXIMUM MESH-SIZE IN REFINEMENT OF
C             VERTICAL MESH,**INPUT, DEFAULT VALUE=1.**
C    HNEW     REAL VECTOR VARIABLE, USED IN VERTICAL MESH REFINEMENT
C             ROUTINE
C    IFMX     INTEGER CONSTANT VARIABLE, MAXIMUM MESH SUB-DIVISION IN
C             VERTICAL MESH REFINEMENT
C       J     INTEGER CONSTANT VARIABLE, NUMBER OF VERTICAL POINTS,
C             (=THE NUMBER OF INTERNAL INTERVALS + 1). J MUST BE SUPPLIED
C             BY USER IN SUBROUTINE NAMED  YMESH IF YSUPLY=.TRUE.,
C             **INPUT, DEFAULT VALUE=101**
C    JMAX     INTEGER CONSTANT VARIABLE, MAXIMUM NUMBER OF VERTICAL
C             POINTS ALLOWED
C      KC     INTEGER CONSTANT VARIABLE, NUMBER OF CONTINUATION IN
C             OBTAINING SOLUTION TO STREAMWISE VELOCITY COMPONENTS AT
C             S=0.**INPUT, DEFAULT VALUE=1**
C   KSAME     INTEGER CONSTANT VARIABLE, USED IN VERTICAL MESH REFINE-
C             MENT. =0 IF THERE IS NO CHANGE IN DISTRIBUTION OF VERTICAL
C             MESH
C    KTER     INTEGER CONSTANT VARIABLE, =0 IF NEWTON≠S METHOD FAILS TO
C             CONVERGE
C    KPRT     INTEGER VECTOR VARIABLE, STREAMWISE STATIONS AT WHICH THE
C             SOLUTION IS TO BE PRINTED ON PAPER, STATEMENT MUST BE ADDED
C             TO SUBROUTINE PRMESH TO INDICATE THE STATIONS IF OPTPT=
C             .TRUE.. THE VECTOR MUST BE SUPPLIED IF XSUPLY=.TRUE.
C   MTEST     INTEGER CONSTANT VARIABLE, CRITERION IN CONVERGENCE TEST
C             OF NEWTON≠S METHOD
C  MAXITS     INTEGER CONSTANT VARIABLE, MAXIMUM NUMBER OF ITERATIONS
C             ALLOWED IN NEWTON≠S METHOD
C      NC     INTEGER VECTOR VARIABLE, ROW PERMUTATION VECTOR USED IN
C             SOLUTION OF BLOCK TRIDIAGONAL SYSTEM
C     NCR     INTEGER VECTOR VARIABLE, PIVOTING STRATEGY INFORMATION
C             VECTOR
C      NR     INTEGER VECTOR VARIABLE, COLUMN PERMUTATION VECTOR USED
C             IN SOLUTION OF BLOCK TRIDIAGONAL SYSTEM
C      NU     INTEGER CONSTANT VARIABLE, NUMBER OF EQUATIONS (=4) OF
C             THE STEAMWISE VELOCITY COMPONENT (F,D/D(ETA) F, D/D(ETA)
C             (D/D(ETA) F)))
C      NW     INTEGER CONSTANT VARIABLE, NUMBER OF EQUATIONS (=2) OF
C             THE SPANWISE VELOCITY COMPONENT (G,D/D(ETA) G)
```

```
C             BE MARCHED. THIS MUST BE SUPPLIED IN SUBROUTINE XMESH IF     00001000
C             XSUPLY=.TRUE.                                                00001010
C   NPRINT    INTEGER CONSTANT VARIABLE, NUMBER OF UNIFORM STREAMWISE      00001020
C             STATIONS AT WHICH SOLUTION IS TO BE PRINTED (XB-XA)/NPRINT.  00001030
C             **INPUT, DEFAULT VALUE=10**                                  00001040
C    OPTPT    LOGICAL VARIABLE, =.TRUE. IF USER IS TO SUPPLY STREAMWISE    00001050
C             STATIONS IS BE PRINTED ON PAPER, **INPUT, DEFAULT VALUE=     00001060
C             .FALSE.**                                                    00001070
C       P1    REAL CONSTANT VARIABLE, (S/H2) * D/DS (H2)                   00001080
C       P2    REAL CONSTANT VARIABLE, RADIUS OF CURVATURE OF CURVED WALL   00001090
C       P4    REAL CONSTANT VARIABLE, D/DS (H2)                            00001100
C       P5    REAL CONSTANT VARIABLE, - (1./H1 / H2) * D/DS ( H1 )         00001110
C       P6    REAL CONSTANT VARIABLE, (S-0.5*HX)/HX                        00001120
C   REFINE    LOGICAL VARIABLE, =.TRUE. IF VERTICAL MESH IS TO BE REFINED.00001130
C             **INPUT, DEFAULT VALUE=.TRUE.**                              00001140
C       UH    REAL VECTOR VARIABLE, USED IN SETUP OF BLOCK TRIDIAGONAL     00001150
C             SYSTEM                                                       00001160
C      UHX    REAL VECTOR VARIABLE, USED IN SETUP OF BLOCK TRIDIAGONAL     00001170
C             SYSTEM                                                       00001180
C       UT    REAL VECTOR VARIABLE, STEAMWISE VELOCITY COMPONENT AT THE    00001190
C             PRESENT STREAMWISE STATION, UT(K,L) HAS THE VALUE OF KTH     00001200
C             COMPONENT OF UT AT ETA = ETA(L), WHERE ETA(1)=YA=0 AND       00001210
C             ETA(M)=ETA(M-1)+H(M-1), M=2,3,...,J-1. THUS UT(1,3) HAS      00001220
C             VALUE OF F AT ETA(3)                                         00001230
C      UTX    REAL VECTOR VARIABLE, STREAMWISE VELOCITY COMPONENT AT THE   00001240
C             PREVIOUS STATION                                             00001250
C       WT    REAL VECOTR VARIABLE, SPANWISE VELOCITY COMPONENTS (G,       00001260
C             D/D(ETA) G) AT THE PRESENT STREAMWISE STATION, SEE UT FOR    00001270
C             STORAGE CONVENTION                                           00001280
C      WTX    REAL VECTOR VARIABLE, SPANWISE VELOCITY VECTOR AT THE        00001290
C             PREVIOUS STREAMWISE STATION                                  00001300
C      XPT    REAL CONSTANT VARIABLE, UNIFORM INTERVAL TO WHICH SOLUTION   00001310
C             IS TO BE PRINTED =(XB-XA)/NPRINT                             00001320
C   XSUPLY    LOGICAL VARIABLE, =.TRUE. IF USER SUPPLIES THE STREAMWISE    00001330
C             MESH.**INPUT, DEFAULT VALUE=.FALSE.**                        00001340
C   YSUPLY    LOIGCAL VARIABLE, =.TRUE. IF USER SUPPLIES THE VERTICAL      00001350
C             MESH,**INPUT, DEFAULT VALUE=.FALSE.**                        00001360
C       XA    REAL CONSTANT VARIABLE, STARTING STREAMWISE STATION, =0      00001370
C       XB    REAL CONSTANT VARIABLE, THE RIGHT END-POINT OF STREAMWISE    00001380
C             MESH, (I.E. THE LAST STREAMWISE STATION),**INPUT, DEFAULT    00001390
C             VALUE=1.**                                                   00001400
C       YA    REAL CONSTANT VARIABLE, THE LEFT END-POINT OF VERTICAL       00001410
C             MESH, =0                                                     00001420
C       YB    REAL CONSTANT VARIABLE. THE RIGHT END-POINT OF VERTICAL      00001430
C             MESH,**INPUT, DEFAULT VALUE=20.**                            00001440
C                                                                          00001450
C                                                                          00001460
C ***********************************************************************  00001470
C *                                                                     *  00001480
C *     INPUT REQUIREMENTS                                              *  00001490
```

```
C ******************************************************************
C
C FOR INPUT, THE FOLLOWING VARIABLES MUST BE SUPPLIED BY USER
C
C     C1,C4,C7,CASEW
C
C WITH THE FOLLOWING OPTIONAL
C
C     XSUPLY,YSUPLY,XH,HKS,FACX,J,REFINE,HMAX,OPTPT,NPRINT,C3,KC
C
C FOR COFLOWING CASES, USER MUST INPUT ARBITRARY FUNCTION (SEE DESCRIP-
C TION OF INTEGER VARIABLE  C4) IN THE SUBROUTINE NAMED BC
C IF THE LOGICAL VARIABLE XSUPLY=.TRUE., USER MUST INPUT STREAMWISE
C MESH AND TOTAL NUMBER OF STREAMWISE STATIONS THROUGH THE SUBROUTINE
C NAMED  XMESH
C IF THE LOGICAL VARIABLE YSUPLY=.TRUE., USER MUST INPUT VERTICAL MESH
C AND TOTAL NUMBER OF VERTICAL POINTS THROUGH SUBROUTINE NAMED  YMESH
C
C
C ******************************************************************
C *                                                                *
C *    TROUBLE SHOOTING AND SOME HINTS IN USING THIS PROGRAM       *
C *                                                                *
C ******************************************************************
C
C IF OSCILLATIONS ARE DETECTED IN THE SOLUTION BETWEEN SUCCESSIVE
C POINTS IN EITHER STREAMWISE OR VERTICAL DIRECTIONS, MESH SHOULD BE
C HALVED IN THAT DIRECTION AND CALCULATIONS SHOULD BE DONE AGAIN TO
C CHECK IF THE OSCILLATION IS TRULY A PHYSICAL PHENOMENON, RATHER
C THAN NUMERICAL STABILITY. MESHES SHOULD BE SO CHOSEN TO ENSURE
C SOLUTIONS AGREE TO NUMERICAL ACCURACY WHEN MESHES ARE HALVED
C IF THE SOLUTION AT ETA=YB DOES NOT QUIET DOWN SUFFICIENTLY, YB SHOULD
C BE INCREASED
C TO ACHIEVE STRICTER CONVERGENCE IN NEWTON,S ITERATION, EPSERR SHOULD
C BE DECREASED (MUST BE GREATER THAN 1.E-13 WITH CDC 6600) AND MAXITS
C INCREASED. THESE CAN BE DONE BY CHANGING THE DATA STATEMENT IN MAIN
C PROGRAM
C IF MORE ACCURATE SOLUTION IS DESIRED, RICHARDSON EXTRAPOLATION CAN
C BE USED TO GET HIGHER ORDER SOLUTION IN BOTH STREAMWISE AND VERTICAL
C DIRECTIONS
C IF MORE POINTS ARE DESIRED, JMAX SHOULD BE INCREASED BY CHANGING DATA
C STATEMENT IN MAIN PROGRAM.THE COMMON STATEMNTS IN THE MAIN PROGRAM.
C HAVE TO BE CHANGED ACCORDINGLY.
C IF USER WISHES TO SUPPLY OWNINITIAL PROFILE, SUCH A PROFILE MUST
C SATISFY WALL-JET ASSUMPTIONS. THE PROGRAM HAS TO BE MODIFIED BY
C CHANGING STATEMENTS IN THE SUBROUTINE NAMED PREP, CARE MUST BE
C TAKEN IN THE COMPONENT CONVENTION THAT WAS EXPLAINED IN VARIABLE
C DESCRIPTION OF U1
C FOR LARGE INITIAL MASS FLUX (C3 GREATER THAN ONE), IT IS RECOMMENDED
C THAT THE NUMBER OF CONTINUATION (KC) SHOULD ALSO BE HIGHER, A USEFUL
```

```
  C3 IS NOT AN INTEGER
C
C
C REFERENCE 1  H.B. KELLER    -  A NEW DIFFERENCE SCHEME FOR PARABOLIC
C                                PROBLEMS, APPEARED IN  NUMERICAL
C                                SOLUTIONS OF PARTIAL DIFFERENTIAL
C                                EQUATIONS II, PP. 327-350, 1971.
C REFERENCE 2  GLAUERT       -  THE WALL JET, JOURNAL OF FLUID
C                                MECHANICS, VOLUME 1, PP.625-643, 1956
C
C
C
      LOGICAL CASEU,CASEW,REL,XSUPLY,YSUPLY,REFINE,OPTPT
      COMMON /WRTE/ XPT,OPTPT,NPRINT
      COMMON /OPTION/ XSUPLY,YSUPLY,REFINE,KC
      COMMON /SOLVE/ CASEU,CASEW
      COMMON /NET/ JMAX,HMAX,IFMX,KSAME
      COMMON /MESHX/ HKX(151) /MESHY/ H(151)
      COMMON /UI/ UI(4,151) /UIX/ UIX(4,151) /DU/ DU(4,151)
      COMMON /WI/ WI(2,151) /WIX/ WIX(2,151) /G/ G(4,151)
      COMMON /A/ A(4,4,151) /B/ B(2,4,151) /C/ C(2,4,151) /F/ F(4,151)
      COMMON /NR/ NR(4,151) /NC/ NC(4,151) /NCR/ NCR(4,151)
      COMMON /SETUP/ UH(4),UHX(4),FF(4),AJA(4,4)
      COMMON /KPRT/ KPRT(151)
      COMMON /HNEW/ HNEW(151)
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /CONST/ C1,C2,C3,C4,C5,C6,C7
      COMMON /PARM3/ MAXITS,EPSERR,KTER,REL,CUTOFF,MTEST
      COMMON /PARM4/ NU,NW,J
      COMMON /PARM5/ FACX,HKS,XA,XB,YA,YB,NX
      COMMON /KPPT/ KPPT(50)
C
      NAMELIST /INPUTS/ CASEW,XSUPLY,YSUPLY,XB,YA,HKS,FACX,J,REFINE,
     1   HMAX,OPTPT,NPRINT,C1,C2,C3,C4,C5,C6,C7,KC
      DATA MAXITS,EPSERR,REL,CUTOFF,MTEST,NU,NW /15,1.E-6,.FALSE.,
     1   1.E-3,5,4,2/
      DATA JMAX /151/
C
C DEFAULT VALUES
C
      OPTPT=.FALSE.
      CASEU=.TRUE.
      CASEW=.TRUE.
      XSUPLY=.FALSE.
      YSUPLY=.FALSE.
      REFINE=.TRUE.
      NPRINT=10
      FACX=1.2
      HKS=1.E-5
      YB=10.
```

```
      HMAX=(YB/3.)/(0.6*(J-1))*3.
      XB=1.
      XPT=XB/NPRINT
      C4= .
      C1=1./3.
      C7=-10.
      C3=1.
      C2=0.
      NC=1
C
C INITIALISE
C
      KSTART=1
      TA=0.
      XA=0.
      KTER=1
C
C VECTOR ** G ** IS SET TO ONE FOR THE LAMINAR CASE
C

      DO 30 L=1,JMAX
      DO 10 K=1,2
      G(K,L)=1.
   10 CONTINUE
      DO 20 K=1,4
      UTX(K,L)=0.
   20 CONTINUE
   30 CONTINUE
      READ(5,INPUTS)
      CALL PREP(KSTART)
      WRITE(6,INPUTS)
      CALL WALJET(KSTART)
      STOP
      END
      SUBROUTINE WALJET(KSTART)
C
C THIS SUBROUTINE COMPUTES THE STREAMWISE DEVELOPMENT OF EITHER OR BOTH
C OF THE STREAMWISE VELOCITY  AND SPANWISE VELOCITY FOR A THREE_
C DIMENSIONAL WALL-JET ON A CURVED SURFACE WITH SMALL CROSS-FLOW
C ASSUMPTION
C
      LOGICAL CASEU,CASEW,OPTPT
      COMMON /WRITE/ XPRT,OPTPT,NPRINT /KPRT/ KPRT(1)
      COMMON /SOLVE/ CASEU,CASEW
      COMMON /MESHX/ HRX(1) /MESHY/ H(1) /UT/ UT(4,1) /UTX/ UTX(4,1)
      COMMON /G/ G(4,1) /WT/ WT(2,1) /WTX/ WTX(2,1)
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM3/ MAXITS,EPSERR,KTER,REL,CUTOFF,MITEST
      COMMON /PARM4/ NU,NW,J
```

```
      COMMON /PARMB/ KK,XN,HX,X
C
      N=4
      NP=2
      NQ=2
      NPW=1
      NQW=1
      KASEU=2
      KASEW=3
      KXNP=1
C
C
C SET UP MESHES AND INITIAL PROFILE
C
      XN=XA
      KI=KSTART+1
C
      DO 2000 KOUNT=KI,NX
      KK=KOUNT
      HX=HKX(KOUNT)
      IF((XN+HX).GT.XB) HX=XB-XN
      XN=XN+HX
      X=XN-HX/2.
      WRITE(6,6000) KOUNT,XN,HX
C
C SOLUTION AT PREVIOUS STREAMWISE STATION IS STRORED IN VECTOR ** UIX **
C AS A FIRST GUESS TO EDDY VISCOSITY ,VALUE AT PREVIOUS STREAMWISE
C STATION IS USED
C
      DO 1140 L=1,J
      DO 1100 K=1,N
 1100 UIX(K,L)=UI(K,L)
 1140 G(1,L)=G(2,L)
      DO 1143 L=1,J
      DO 1143 K=1,NW
 1143 WI(K,L)=0.
C
C COMPUTE STREAMWISE MOMENTUM EQUATION AT THE PRESENT STATION
C
      CALL NEWTON(KASEU,NU,NP,NQ)
C
      IF((.NOT.CASEW).OR.(KTER.EQ.0)) GO TO 1159
C
      DO 1156 L=1,J
      DO 1156 K=1,NW
      WIX(K,L)=WI(K,L)
 1156 WI(K,L)=0.
C
C COMPUTE SPANWISE MOMENTUM EQUATION SOLUTION
C
```

```
C
 1159 CONTINUE
C
C EXIT IF NO CONVERGENCE
C
      IF(KTER.EQ.0) GO TO 2100
C
      IF(KOUNT.LT.5) GO TO 1250
      IF(.NOT.OPTPT.AND.(XN.GE.KXNP*XPRT)) GO TO 1200
      IF(OPTPT.AND.(KOUNT.EQ.KPRT(KXNP))) GO TO 1200
      GO TO 1300
 1200 CONTINUE
      KXNP=KXNP+1
 1250 CONTINUE
C
C WRITE SOLUTION ON PAPER
C
      CALL OUTPT(J,YA)
 1300 CONTINUE
C
C
      IF(UT(3,1).GE.0.) GO TO 2050
C
C SEPARATION, THIRD DERIVATIVE OF F (STREAM-FUNCTION OF STREAMWISE
C VELOCITY) IS NEGATIVE
C
      WRITE(6,6200)
      CALL OUTPT(J,YA)
      GO TO 2100
 2050 CONTINUE
      IF(XN.GE.XB) GO TO 2100
 2000 CONTINUE
 2100 CONTINUE
C
 6000 FORMAT(* STATION = *,I4,*, X =*,E12.5,*, X-STEP =*,E12.5)
 6100 FORMAT(* O(H2) SOLUTION*)
 6200 FORMAT(///*XXXXXXXXXXXXXX SEPARATION XXXXXXXXXXXXXX*///)
C
      RETURN
      END
      SUBROUTINE PREP(KSTART)
C
C THIS SUBROUTINE SETS UP THE MESHES FOR BOTH THE STREAMWSIE AND
C VERTICAL DIRECTIONS, AND COMPUTES THE INITIAL SIMILAR SOLUTION FOR A
C GIVEN MASS FLUX
C
      LOGICAL XSUPLY,YSUPLY,REFINE,OPTPT
      COMMON /OPTION/ XSUPLY,YSUPLY,REFINE,KC
      COMMON /WRITE/ XPT,OPTPT,NPRINT
      COMMON /CONST/ C1,C2,C3,C4,C5,C6,C7
```

```
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM3/ MAXITS,EPSERR,KTER,REL,CUTOFF,MTEST
      COMMON /PARM4/ NU,NW,J
      COMMON /PARM5/ FACX,HKS,XA,XB,YA,YB,NX
      COMMON /PARM6/ KOUNT,XN,HX,X
      COMMON /NET/ JMAX,HMAX,IFMX,KSAME
      COMMON /UT/ UT(4,1) /DU/ DU(4,1) /F/ F(4,1) /UTX/ UTX(4,1)
C
      F1(T)=(1.-EXP(-T))/(1.+EXP(-T))
      F2(T)=2.*EXP(-T)/(1.+EXP(-T))**2
      F3(T)=-2.*EXP(-T)/(1.+EXP(-T))**2
      F4(T)=0.
C
C STREAMWISE MESH
C
      IF(.NOT.XSUPLY) GO TO 90
      CALL XMESH
      GO TO 150
   90 CONTINUE
      HKX(1)=HKS
      KT=
      KL=1
      XT=-HKS
C
  100 CONTINUE
      IF((KL.GT.100).OR.(XT.GT.XB)) GO TO 140
      XT=XT+HKX(KL)
      IF(HKX(KL).GT.XPT) FACX=1.
      IF(XT.GE.(KT*XPT)) GO TO 110
      HKX(KL+1)=HKX(KL)*FACX
      KL=KL+1
      GO TO 100
  110 CONTINUE
      IF(FACX.EQ.1.) GO TO 130
      IF((XT-KT*XPT).GT.(HKX(KL)/20.)) GO TO 120
      HKX(KL+1)=HKX(KL)*FACX+(XT-KT*XPT)
      HKX(KL)=KT*XPT-(XT-HKX(KL))
      XT=KT*XPT
      KL=KL+1
      KT=KT+1
      GO TO 100
  120 CONTINUE
      HKX(KL+2)=HKX(KL)*FACX
      HKX(KL+1)=XT-KT*XPT
      HKX(KL)=KT*XPT-(XT-HKX(KL))
      KL=KL+2
      KT=KT+1
      GO TO 100
  130 CONTINUE
      HKX(KL)=KT*XPT-(XT-HKX(KL))
```

```
  135 CONTINUE
      HNX(KE+1)=XPT
      KT=KT+1
      KL=KL+1
      XT=XT+HNX(KL)
      IF(XT.LE.XH) GO TO 135
  140 CONTINUE
C
      NX=KL-1
  150 CONTINUE
C
      IF(OPTPT) CALL PRMESH
C
C VERTICAL MESH
C
      IF(.NOT.YSUPLY) GO TO 350
      CALL YMESH
      GO TO 400
  350 CONTINUE
      J1=J-1
      JA=(J-1)*3./5.
      JB=JA+1
      JC=J-1-JA
      H1=YB/3./JA
      DO 160 L=1,JA
      H(L)=H1
  160 CONTINUE
      H1=2.*YB/3./JC
      DO 170 L=JB,J1
      H(L)=H1
  170 CONTINUE
  400 CONTINUE
C
      J1=J-1
C
C SOLVES FOR INITIAL SIMILAR SOLUTION
C
      IF(KSTART.GT.1) RETURN
      Y=0
      H(J)=0.
      DO 190 L=1,J
      DO 190 K=1,NU
      UIX(K,L)=0.
  190 CONTINUE
      DO 200 L=1,J
      UI(1,L)=F1(Y)
      UI(2,L)=F2(Y)
      UI(3,L)=F3(Y)
      UI(4,L)=F4(Y)
      Y=Y+H(L)
```

```
  200 CONTINUE
      YB=Y
C
C DUMMY INITIALISATION
C
      XN=1.
      HX=1.
      KOUNT=1
      N=4
      NP=2
      NQ=2
      X=0.
C
      C3SAVE=C3
      C3=0.
      DC3=C3SAVE/KC
C
C CONTINUATION PROCEDURE TO OBTAIN INITIAL PROFILE FOR MASS FLUX = C3
C
      DO 2400 KCONT=1,KC
      C3=C3+DC3
      CALL NEWTON(2,N,NP,NQ)
      IF(.NOT.REFINE) GO TO 2100
      CALL NETRUN
      IF(KSAME.EQ.0) WRITE(6,6000)
C
C RECOMPUTE SOLUTION ON THE REFINED NET
C
      IF(KSAME.EQ.1) CALL NEWTON(2,N,NP,NQ)
 2100 CONTINUE
C
C WRITE SOLUTION ON PAPER FOR INITIAL MASS FLUX = C3, WHERE C3
C = KCONT * DC3
C
      CALL OUTPT(J,YA)
      IF(KTER.EQ.0) STOP
      IF(KCONT.EQ.KC) GO TO 2500
C
C OBTAIN IMPROVED INITIAL GUESS FOR NEXT C3 BY SOLVING FIRST
C VARIATIONAL EQUATION
C
      DO 2200 L=1,J
      DO 2200 K=1,NQ
 2200 F(K,L)=0.
      F(3,J)=1.+2.*(UT(2,J)+UT(3,J)/C3)/C3**2
      CALL BLOCK2
      DO 2300 L=1,J
      DO 2300 K=1,NQ
 2300 UT(K,L)=UT(K,L)+DU(K,L)
 2400 CONTINUE
 2500 CONTINUE
```

```
      [illegible] - NO CHANGE IN REFINEMENT*)
C
      RETURN
      END
      SUBROUTINE NEWTON(KASE,N1,N2,N3)
C
C THIS SUBROUTINE SOLVES FOR THE STREAMWISE MOMENTUM EQUATION BY NEWTON
C ITERATION, IF KASE = 2. IF KASE = 3,THE SPANWISE MOMENTUM EQUATION
C IS SOLVED. NEWTON ITERATION IS NOT REQUIRED BECAUSE EQUATION IS
C LINEAR. A CONSEQUENCE OF SMALL CROSS-FLOW ASSUMPTION.
C
      LOGICAL REL
      COMMON /PARM1/ N,NP,NQ
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM3/ MAXITS,EPSERR,KTER,REL,CUTOFF,MTEST
      COMMON /PARM4/ NU,NW,J
      COMMON /PARM6/ KOUNT,XN,HX,X
      COMMON /WT/ WT(2,1) /UT/ UT(4,1) /UU/ UU(4,1)
      COMMON /PREPBP/ IFLAG,YIC,YTOLD,YTOLD2,ITER,YICC
      EXTERNAL JACOB,RHSF,BC,RHSFW,BCW,JACOBW
C
C INITIALISE
C
      N=N1
      NP=N2
      NW=N3
      P6=(XN-.5*HX)/HX
      IF(KOUNT.EQ.1) P6=0.
      IFLAG=1
      YIC=0.
      YTOLD=0.
C
C CONTRIBUTION FROM PREVIOUS STATION FOR THE STREAMWISE MOMENTUM
C EQUATION
C
      IF(KASE.EQ.2) CALL PREPG
C
C CONTRIBUTION FROM PREVIOUS STATION FOR THE SPANWISE MOMENTUM EQUATION
C
      IF(KASE.EQ.3) CALL PREPGW
C
C NEWTON ITERATION
C
      DO 300 M=1,MAXITS
      ITER=M
C
C SOLVE FOR STREAMWISE MOMENTUM EQUATION
C
      IF(KASE.EQ.2) CALL BOX(KASE,RHSF,JACOB,BC)
C
```

```
C
      IF(KASE.EQ.3) CALL BOX(KASE,RHSFW,JACOBW,BCW)
C
C CALL BLOCK TRIDIAGONAL MATRIX SOLVER
C
      CALL BLOCK1
      CALL BLOCK2
C
      IF(KASE.EQ.2) GO TO 190
C
C SPANWISE SOLUTION
C
      DO 180 L=1,J
      DO 180 K=1,N
  180 WT(K,L)=DU(K,L)
      RETURN
C
  190 CONTINUE
C
C ERROR TESTING BETWEEEN TWO NEWTON ITERATES
C
      ERMAX=0.
      DO 200 L=1,J
      DO 200 K=1,N
      UT(K,L)=UT(K,L)+DU(K,L)
      ERROR=ABS(DU(K,L))
C
C ABSOLUTE ERROR CRITERION IS USED IF VARIABLE ** REL ** IS SET TO
C .FALSE.
C
      IF(.NOT.REL) GO TO 195
      IF(ABS(UT(K,L)).GT.1.E_4) ERROR=ERROR/ABS(UT(K,L)+0.5*DU(K,L))
  195 CONTINUE
      IF(ERMAX.GE.ERROR) GO TO 200
      LOCX=K
      LOCY=L
      ERMAX=ERROR
  200 CONTINUE
C
      IF(ERMAX.GE.EPSERR) GO TO 205
      WRITE(6,6000) M,ERMAX,LOCX,LOCY,YTCC,UT(3,1),UT(4,1)
      RETURN
  205 CONTINUE
      IF(M.LT.MIFST) GO TO 210
      IF((ERMAX.LT.CUTOFF).AND.(EROLD/ERMAX.LT.1.05)) RETURN
  210 CONTINUE
      EROLD=ERMAX
C
      IF(KOUNT.GT.1) CALL PREPB
  300 CONTINUE
```

```
C NO CONVERGENCE. WRITE FINAL ITERATE AND EXIT
C
      KITER=0
      WRITE(6,6100)
      CALL OUTPT(J,YA)
C
 6000 FORMAT(* NEWTON - ITER =*,I2,* ERROR =*,E12.5,* AT(*,I2,*,*,I3,
     1  * )*,5X,* YTC = *,E12.5,* U(3,1, *,E12.5,* P *,E12.5)
 6100 FORMAT(* NEWTON - NO CONVERGENCE IN NEWTON ITERATION *)
C
      RETURN
      END
      SUBROUTINE BOX(KASE,RHSF,JACOB,BC)
C
C THIS SUBROUTINE SETS UP THE BLOCK TRIDIAGONAL SYSTEMS OF EQUATIONS FOR
C (A) STREAMWSIE MOMENIUM EQUATION WHEN KASE = 2, AND (B) SPANWISE
C MOMENIUM EQUATION WHEN KASE = 3
C EQUATION WHEN KASE = 3
C
      COMMON /A/ A(4,4,1) /B/ B(2,4,1) /C/ C(2,4,1) /F/ F(4,1)
      COMMON /MESHY/ H(1) /UT/ U(4,1) /UTX/ UTXN(4,1) /G/ G(4,1)
      COMMON /WT/ WT(2,1) /WTX/ WTX(2,1)
      COMMON /SETUP/ UH(4),UHXN(4),FF(4),AJA(4,4)
      COMMON /PARM1/ N,NP,NQ
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM4/ NU,NW,J
      COMMON /PARM5/ FACX,HKS,XA,XB,YA,YB,NA
      COMMON /PARM6/ KOUNT,XN,HX,X
C
      J1=J-1
      DO 30 M=1,J
      DO 30 L=1,N
      DO 10 K=1,NP
      A(K,L,M)=0.
   10 B(K,L,M)=0.
      DO   0 K=1,NQ
      C(K,L,M)=0.
   20 A(K+NP,L,M)=0.
   30 CONTINUE
C
      DO 40 L=1,N
      DO 40 K=1,N
   40 AJA(K,L)=0.
      DO 50 K=1,N
      UH(K)=U(K,1)
   50 UHXN(K)=U(K,J)
C
C BOUNDARY CONDITIONS CONTRIBUTION
C
      CALL BC
```

```
      DO 80 L=1,N
   80 A(K,L,1)=AJA(K,L)
   90 F(K,1)=-FF(K)
      DO 200 K=1,NQ
      DO 100 L=1,N
  100 A(K+NP,L,J)=AJA(K+NP,L)
  200 F(K+NP,J)=-FF(K+NP)
C
      DO 210 L=1,N
      DO 210 K=1,N
  210 AJA(K,L)=0.
C
C INTERNAL POINTS CONTRIBUTION
C
      Y=YA+H(1)/2.
      DO 900 M=2,J
      M1=M-1
      CALL PREPP(Y)
      DO 300 K=1,N
      UHXN(K)=(UTXN(K,M1)+UTXN(K,M))/2.
  300 UH(K)=(U(K,M1)+U(K,M))/2.
      CALL JACOB
      CALL RHSF
      DO 600 K=1,NQ
      KP=K+NP
      DO 500 L=1,N
      HA=-AJA(K,L)*H(M1)/2.
      A(KP,L,M1)=HA
  500 C(K,L,M1)=HA
      F(KP,M1)=U(K,M1)-U(K,M)+H(M1)*FF(K)
      A(KP,K,M1)=-1.+A(KP,K,M1)
  600 C(K,K,M1)=1.+C(K,K,M1)
      DO 800 K=1,NP
      KQ=K+NQ
      DO 700 L=1,N
      HA=-AJA(KQ,L)*H(M1)/2.
      A(K,L,M)=HA
  700 B(K,L,M)=HA
      F(K,M)=U(KQ,M1)-U(KQ,M)+H(M1)*FF(KQ)
      A(K,KQ,M)=1.+A(K,KQ,M)
  800 B(K,KQ,M)=-1.+B(K,KQ,M)
C
C CONTRIBUTION FROM PREVIOUS STREAMWISE STATION
C
      IF(KASE.EQ.3) GO TO 850
      B(1,2,M)=B(1,2,M)-P3*G(2,M1)
      B(1,3,M)=B(1,3,M)+1.-G(2,M1)
      A(1,2,M)=A(1,2,M)+P3*G(2,M)
      A(1,3,M)=A(1,3,M)-1.+G(2,M)
      F(1,M)=F(1,M)+U(3,M)-U(3,M1)-G(3,M)+G(2,M1)*(U(3,M1)+P3*U(2,M1))
```

```
      F(2,M)=F(2,M)-G(4,M)
      GO TO 860
  850 CONTINUE
      B(1,1,M)=B(1,1,M)-P3*G(2,M1)
      B(1,2,M)=B(1,2,M)+1.-G(2,M1)
      A(1,1,M)=A(1,1,M)-P3*G(2,M)
      A(1,2,M)=A(1,2,M)-1.+G(2,M)
      F(2,M1)=0.
      F(1,M)=-G(3,M)
  860 CONTINUE
      Y=Y+(H(M1)+H(M))/2.
  900 CONTINUE
C
C REARRANGE EQUATIONS TO ENSURE THE FIRST DIAGONAL BLOCK IS NONSINGULAR
C THIS IS CRUCIAL FOR THE BLOCK TRIDIAGONAL SOLVER
C
      IF(KASE.EQ.3) GO TO 960
      LT=2
      KSWICH=1
      K=3
      GO TO 970
  960 CONTINUE
      LT=1
      KSWICH=1
      K=2
  970 CONTINUE
      DO 1500 LL=1,LT
      IF(LL.EQ.1) GO TO 1200
      KSWICH=2
      K=4
 1200 CONTINUE
      DO 1400 M=1,J1
      DO 1300 L=1,N
      T=A(NP+KSWICH,L,M)
      A(NP+KSWICH,L,M)=B(K-NQ,L,M+1)
      B(K-NQ,L,M+1)=T
      T=C(KSWICH,L,M)
      C(KSWICH,L,M)=A(K-NQ,L,M+1)
      A(K-NQ,L,M+1)=T
 1300 CONTINUE
      T=F(NP+KSWICH,M)
      F(NP+KSWICH,M)=F(K-NQ,M+1)
      F(K-NQ,M+1)=T
 1400 CONTINUE
 1500 CONTINUE
C
      RETURN
      END
      SUBROUTINE BLOCKI
C
```

```
C THIS SUBROUTINE DECOMPOSES THE BLOCK TRIDIAGONAL MATRIX INTO LU-FORM
C
      COMMON /A/ A(4,4,1) /B/ B(2,4,1) /C/ C(2,4,1)
      COMMON /NR/ NR(4,1) /NC/ NC(4,1) /NCR/ NCR(4,1)
      COMMON /PARM1/ N,NP,NQ
      COMMON /PARM4/ NU,NW,J
      DO 100 L=1,J
      DO 100 K=1,N
      NR(K,L)=K
      NC(K,L)=K
  100 NCR(K,L)=0
      CALL LUSOLV(1)
C
      DO 600 M=2,J
      M1=M-1
      KM=M
C
      CALL BETASV(KM)
C
C SOLVE SCALAR MATRIX ALPHA
C
      DO 500 K=1,NP
      DO 400 L=1,N
      SUM=0.
      DO 200 KK=1,NQ
  200 SUM=SUM-B(K,KK+NP,M)*C(KK,L,M))
      A(K,L,M)=A(K,L,M)+SUM
  400 CONTINUE
  500 CONTINUE
C
      CALL LUSOLV(KM)
C
C
  600 CONTINUE
C
      RETURN
      END
      SUBROUTINE LUSOLV(KM)
C
C THIS SUBROUTINE DECOMPOSES A SECALAR MATRIX INTO LU-FORM USING
C A MIXED PIVOTING STRATEGY
C
      COMMON /NR/ NR(4,1) /NC/ NC(4,1) /NCR/ NCR(4,1) /A/ A(4,4,1)
      COMMON /PARM1/ N,NP,NQ
C
      DO 500 M=2,N
      M1=M-1
C
C SEARCH FOR OPTIMAL PIVOT
C
      KQ=M1
```

```
      NRM=NR(MI,KM)
      NCM=NC(MI,KM)
      CPIVOT=A(NRM,NCM,KM)
      RPIVOT=CPIVOT
      DO 200 K=M,N
      NRK=NR(K,KM)
      NCK=NC(K,KM)
      IF(ABS(RPIVOT).GE.ABS(A(NRK,NCM,KM))) GO TO 100
C
      KR=K
      RPIVOT=A(NRK,NCM,KM)
  100 CONTINUE
      IF(ABS(CPIVOT).GE.ABS(A(NRM,NCK,KM))) GO TO 200
      KC=K
      CPIVOT=A(NRM,NCK,KM)
  200 CONTINUE
      IF(ABS(RPIVOT).GE.ABS(CPIVOT)) GO TO 400
C
C PIVOT BY INTERCHANGING COLUMN
C
      IF(ABS(CPIVOT).LT.1.E-10) WRITE(6,6000) KM,MI,CPIVOT
      NCR(MI,KM)=1
      KI=NC(KC,KM)
      IF(KC.NE.MI) KSIGN=KSIGN+1
      NC(KC,KM)=NC(MI,KM)
      NC(MI,KM)=KI
C
C GAUSSIAN ELIMINATION
C
      DO 300 L=M,N
      NCL=NC(L,KM)
      A(NRM,NCL,KM)=A(NRM,NCL,KM)/CPIVOT
      T=A(NRM,NCL,KM)
      DO 300 K=M,N
      NRK=NR(K,KM)
  300 A(NRK,NCL,KM)=A(NRK,NCL,KM)-T*A(NRK,KI,KM)
      GO TO 600
C
C PIVOT BY INTERCHANGING ROW
C
  400 CONTINUE
      IF(ABS(RPIVOT).LT.1.E-10) WRITE(6,5000) KM,MI,RPIVOT
      KI=NR(KR,KM)
      IF(KR.NE.MI) KSIGN=KSIGN+1
      NR(KR,KM)=NR(MI,KM)
      NR(MI,KM)=KI
C
C GAUSSIAN ELIMINATION
C
      DO 500 L=M,N
```

```
      A(NRL,NCM,KM)=A(NRL,NCM,KM)/RPIVOT
      T=A(NRL,NCM,KM)
      DO 500 K=M,N
      NCK=NC(K,KM)
  500 A(NRL,NCK,KM)=A(NRL,NCK,KM)-T*A(KT,NCK,KM)
C
  600 CONTINUE
      NRN=NR(N,KM)
      NCN=NC(N,KM)
      IF(ABS(A(NRN,NCN,KM)).LT.1.E-10) WRITE(6,6100) KM,A(NRN,NCN,KM)
 6000 FORMAT(* LUSOLV - BLOCK =*,I4,* PIVOT AT EQN NO.=*,I2,* IS*,E12.5)
 6100 FORMAT(* LUSOLV - BLOCK =*,I4,* LAST PIVOT IS*,E12.5)
C
      RETURN
      END
      SUBROUTINE BETASV(KM)
C
C THIS SUBROUTINE SOLVES FOR BETA IN THE LU-DECOMPOSITION OF THE BLOCK
C TRIDIAGONAL MATRIX
C
      COMMON /SETUP/ UH(4),UHX(4),FF(4),AJA(4,4)
      COMMON /A/ A(4,4,1) /B/ B(2,4,1) /C/ C(2,4,1)
      COMMON /NR/ NR(4,1) /NC/ NC(4,1) /NCR/ NCR(4,1)
      COMMON /PARM1/ N,NP,NW
      COMMON /PARM4/ NU,NW,J
C
C SOLVE Y IN Y * (U*Q)  =  B
C
      M1=KM-1
      DO 700 M=1,NP
      DO 300 L=1,N
      NCL=NC(L,M1)
      SUM=B(M,NCL,KM)
      IF(L.EQ.1) GO TO 200
      L1=L-1
      DO 100 K=1,L1
      NRK=NR(K,M1)
  100 SUM=SUM-UH(K)*A(NRK,NCL,M1)
  200 UH(L)=SUM
      NRL=NR(L,M1)
      IF(NCR(L,M1).EQ.0) UH(L)=UH(L)/A(NRL,NCL,M1)
  300 CONTINUE
C
C SOLVE BETA IN BETA * A  =  B
C
      DO 500 LL=2,N
      L=N-LL+1
      L1=L+1
      SUM=UH(L)
      NCL=NC(L,M1)
```

```
      NRK=NR(K,M1)
  400 SUM=SUM-UH(K)*A(NRK,NCL,M1)
      UH(L)=SUM
      NRL=NR(L,M1)
      IF(NCR(L,M1).EQ.1) UH(L)=UH(L)/A(NRL,NCL,M1)
  500 CONTINUE
C
C
C REARRANGE COMPONENTS DUE TO MIXED PIVOTING
C
      DO 600 L=1,N
      NRL=NR(L,M1)
  600 B(M,NRL,K4)=UH(L)
  700 CONTINUE
C
      RETURN
      END
      SUBROUTINE BLOCK2
C
C ASSUMING THE BLOCK TRIDIAGONAL MATRIX IS IN FACTORISED FORM, THIS
C ROUTINE COMPUTES THE SOLUTION FOR A PARTICULAR RIGHT HAND SIDE
C
      COMMON /B/ B(2,4,1) /C/ C(2,4,1) /F/ F(4,1) /DU/ DU(4,1)
      COMMON /PARM1/ N,NP,NQ
      COMMON /PARM4/ NU,NW,J
C
C SOLVE Y IN L * Y  =  F
C
      DO 300 M=2,J
      M1=M-1
      DO 200 K=1,NP
      SUM=0.
      DO 100 KK=1,N
  100 SUM=SUM-B(K,KK,M)*F(KK,M1)
      F(K,M)=F(K,M)+SUM
  200 CONTINUE
  300 CONTINUE
C
C SOLVE DU IN A * DU  =  F
C
      CALL USOLVF(J)
C
      DO 600 MM=2,J
C
C UPDATE RIGHT HAND SIDE
C
      M1=J-MM+2
      M=M1-1
      DO 500 K=1,NQ
      SUM=0.
```

```
      DO 400 L=1,N
  400 SUM=SUM-C(K,L,M)*DU(L,M))
  500 F(NP+K,M)=F(NP+K,M)+SUM
C
      CALL USOLVE(M)
C
  600 CONTINUE
C
      RETURN
      END
      SUBROUTINE USOLVE(M)
C
C ASSUMMING A SCALAR MATRIX IS IN FACTORISED FORM, THIS ROUTINE SOLVES
C THE SOLUTION FOR A PARTICULAR RIGHT HAND SIDE
C
      COMMON /A/ A(4,4,1) /F/ F(4,1) /DU/ DU(4,1)
      COMMON /NR/ NR(4,1) /NC/ NC(4,1) /NCR/ NCR(4,1)
      COMMON /PARM1/ N,NP,NQ
C
C SOLVE Y IN L * Y  =  F
C
      DO 300 L=1,N
      NRL=NR(L,M)
      SUM=F(NRL,M)
      IF(L.EQ.1) GO TO 200
      L1=L-1
      DO 100 K=1,L1
      NCK=NC(K,M)
  100 SUM=SUM-A(NRL,NCK,M)*DU(K,M)
  200 DU(L,M)=SUM
      NCL=NC(L,M)
      IF(NCR(L,M).EQ.1) DU(L,M)=DU(L,M)/A(NRL,NCL,M)
  300 CONTINUE
      NRN=NR(N,M)
      NCN=NC(N,M)
      F(N,M)=DU(N,M)/A(NRN,NCN,M)
C
C
C SOLVE DU IN A * DU  =  F
C
      DO 500 LL=2,N
      L=N-LL+1
      L1=L+1
      SUM=DU(L,M)
      NRL=NR(L,M)
      DO 400 K=L1,N
      NCK=NC(K,M)
  400 SUM=SUM-A(NRL,NCK,M)*F(K,M)
      F(L,M)=SUM
      NCL=NC(L,M)
      IF(NCR(L,M).EQ.0) F(L,M)=F(L,M)/A(NRL,NCL,M)
```

```
C
C
C REARRANGE COMPONENTS DUE TO MIXED PIVOTING
C
      DO 600 L=1,N
      NCL=NC(L,M)
  600 DU(NCL,M)=F(L,M)
      RETURN
      END
      SUBROUTINE PREPP(Y)
C
C P2 IS FOR LOG. SPRIAL, SHOULD BE CHANGED FOR OTHER CURVED WALLS
C
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM6/ KOUNT,XN,HX,X
      COMMON /CONST/ C1,C2,C3,C4,C5,C6,C7
C
      P1=0.
      P2=C1/(X+1.2*C1)
      P3=4.*P2*X**0.75*C2
      P4=0.
      P5=C7
      RETURN
      END
      SUBROUTINE PREPB
C
C RAMAPRIAN TRUBULENCE MODEL ( REFERENCE RAMAPRIAN - TURBULENT WALL-JETS
C DIFFUSERS, AIAA VOL. 11, NO. 12, PP. 1684_1690, 1973 )
C
      COMMON /W1/ W1(2,1) /U1/ U(4,1) /G/ G(4,1) /MESHY/ H(1)
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM4/ NU,NW,J
      COMMON /PARM5/ FACX,HKS,XA,XB,YA,YB,NA
      COMMON /PARM6/ KOUNT,XN,HX,X
      COMMON /PREPBP/ IFLAG,YIC,YIOLD,YIOLD2,ITER,YICC
C
C U HAS THE SAME MEANING AS UI IN THE MAIN PROGRAM
C
      IF(ITER.LT.5) GO TO 90
      IF(IFLAG.EQ.0) GO TO 400
      IF(((YIC-YIOLD).EQ.(-(YIOLD-YIOLD2))).AND.(YIOLD.NE.YIC)) IFLAG=0
      IF(IFLAG.EQ.1) GO TO 90
      YIC=AMAX1(YIC,YIOLD)
      GO TO 400
   90 CONTINUE
      YIOLD2=YIOLD
      YIOLD=YIC
      YI=YB
      J1=J-1
C
```

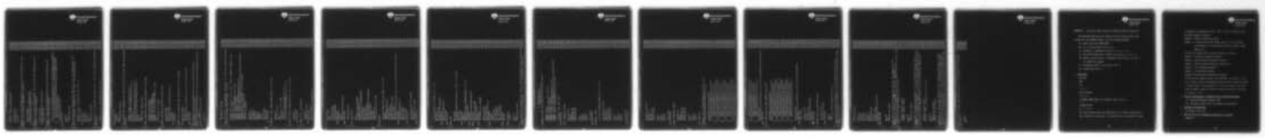
```
C DETERMINE POINT WHERE SECOND LAYER STARTS
C
      DO 100 LL=1,J1
      L=J-LL
      YT=YT-H(L)
      UNORM=SQRT(U(2,L)**2+WT(1,L)**2)
      IF(UNORM.LT.0.01) GO TO 100
      GO TO 200
  100 CONTINUE
  200 CONTINUE
      YTC=YT*.125/.435
      YTCC=YTC
  400 CONTINUE
C
      Y=0.
      DO 300 L=1,J
      UNORM=SQRT(U(3,L)**2+WT(2,L)**2)
C
C FIRST LAYER
C
C
      G(2,L)=1.+X**0.25*(0.435*Y)**2*UNORM
C
C SECOND LAYER
C
      IF(Y.GE.YTC) G(2,L)=1.+X**0.25*(0.125*YT)**2*UNORM
      Y=Y+H(L)
  300 CONTINUE
C
      RETURN
      END
      SUBROUTINE PREPG
C
C THIS SUBROUTINE COMPUTES THE CONTRIBUTION FROM THE PREIVOUS STREAMWISE
C STATION OF THE STREAMWISE MOMENTUM EQUATION. HERE U IS SOLUTION OF
C STREAMWISE VELOCITY VECTOR AT THE PREVIOUS STATION
C
      COMMON /MESHY/ HY(1) /UTX/ U(4,1) /G/ G(4,1)
      COMMON /SETUP/ UH(4),UHX(4),FF(4),AJA(4,4)
      COMMON /PARM2/ P1,P2,P3,P4,P5,P5
      COMMON /PARM4/ N,NW,J
      COMMON /PARM5/ FACX,HKS,XA,XB,YA,YB,HA
      COMMON /PARM6/ KOUNT,XN,HA,X
C
      DECOU=0.
      Y=YA+HY(1)/2.
      DO 100 L=2,J
      L1=L-1
      CALL PREPP(Y)
C
C COMPUTE U AT POINT MIDWAY BETWEEN TWO VERTICAL POINTS AT PREVIOUS
```

```
C
      DO 10 K=1,N
   10 UH(K)=(U(K,L)+U(K,L-1))/2.
C
C COMPUTE CONTRIBUTION
C
      G(3,L)=G(1,L)*(U(3,L)+P3*U(2,L))-G(1,L1)*(U(3,L1)+P3*U(2,L1))
     1  +HY(L-1)*(UH(1)*UH(3)*(1.+P1-4.*P6)+2.*UH(2)**2*(1.+2.*P6)
     2    +4.*X**.75*UH(4)*(8.*P6+1.)*DECOU)
      G(4,L)=U(4,L)-U(4,L1)-HY(L1)*P2*UH(2)**2
      Y=Y+(HY(L-1)+HY(L))/2.
  100 CONTINUE
C
      RETURN
      END
      SUBROUTINE BC
C
C THIS SUBROUTINE COMPUTES THE JACOBIAN MATRIX AND RIGHT HAND SIDE OF
C THE BOUNDARY CONDITIONS FOR THE STREAMWSIE MOMENTUM EQUATION
C FUNCTION STATEMENT FN(X) HAS TO BE INPUT BY USER FOR HIS OWN CO-
C FLOWING BOUNDARY CONDITON AT INFINITY
C
      COMMON /SETUP/ UA(4),UB(4),G(4),B(4,4)
      COMMON /CONST/ C1,C2,C3,C4,C5,C6,C7
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM6/ KOUNT,XN,HX,X
C
      FN(X)=X*EXP(-X)
C
      G(1)=UA(1)
      G(2)=UA(2)
      G(4)=UB(4)
      B(1,1)=1.
      B(2,2)=1.
      B(4,4)=1.
      IF(KOUNT.GT.1) GO TO 100
C
C ASYMPTOTIC BOUNDARY CONDITION AT INFINITY AT S=0
C
      G(3)=UB(1)+2.*UB(2)/C3+UB(3)/C3**2-C3
      B(3,2)=2./C3
      B(3,3)=1./C3**2
      B(3,1)=1.
      RETURN
  100 CONTINUE
C
C BOUNDARY CONDITION FOR S GREATER THAN ZERO
C
      G(3)=UB(2)-SQRT(XN)*C4*FN(XN)
      B(3,2)=1.
```

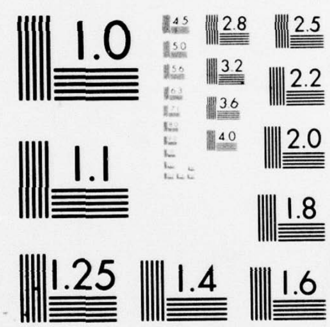
```
C
      RETURN
      END
      SUBROUTINE RHSF
C
C THIS SUBROUTINE COMPUTES THE RIGHT HAND SIDE OF THE STREAMWISE
C MOMENTUM EQUATION
C
      COMMON /SETUP/ U(4),UX(4),F(4),A(4,4)
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM6/ KOUNT,XN,HX,X
C
      DECOU=0.
      F(1 =U(2)
      F(2)=U(3)
      F(3)=-(1.+P1)*U(1)*U(3)+2.*U(2)**2*(2.*P6-1.)-4.*P6*((U(3)+UX(3))
     1   *U(1)-UX(1)*U(3))+4.*DECOU*X**.75*U(4)*(8.*P6-1.)
      F(4)=P2*U(2)**2
C
      RETURN
      END
      SUBROUTINE JACOB
C
C THIS SUBROUTINE COMPUTES THE JACOBIAN MATRIX OF THE STREAMWISE
C MOMENTUM EQUATION
C
      COMMON /SETUP/ U(4),UX(4),F(4),A(4,4)
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM6/ KOUNT,XN,HX,X
C
      DECOU=0.
      A(1,2)=1.
      A(2,3)=1.
      A(3,1)=-(1.+P1)*U(3)-4.*P6*(U(3)+UX(3))
      A(3,2)=4.*U(2)*(-1.+2.*P6)
      A(3,3)=-(1.+P1)*U(1)-4.*P6*(U(1)-UX(1))
      A(3,4)=4.*DECOU*X**.75*(8.*P6-1.)
      A(4,2)=2.*P2*U(2)
C
      RETURN
      END
      SUBROUTINE PREPGW
C
C THIS SUBROUTINE COMPUTES THE CONTRIBUTION FROM THE PREVIOUS STREAMWISE
C STATION OF THE SPANWISE MOMENTUM EQUATION
C
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /PARM4/ NU,NW,J
      COMMON /PARM5/ FACX,HKS,XA,XB,YA,YB,NX
      COMMON /WI/ WI(2,1) /WIX/ WIX(2,1) /UI/ UI(4,1) /UIX/ UIX(4,1)
      COMMON /G/ G(4,1) /MESHY/ HY(1)
```

AD-A048 450 ROCKWELL INTERNATIONAL THOUSAND OAKS CALIF SCIENCE --ETC F/G 20/4
A COMPUTATIONAL MODEL FOR THREE-DIMENSIONAL INCOMPRESSIBLE SMAL--ETC(U)
DEC 77 N D MALMUTH, R K SZETO N62269-76-C-0382
UNCLASSIFIED SC5079.5FR NADC-76-105-30 NL

2 OF 2
ADA048450



END
DATE
FILMED
2-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
      COMMON /PARM6/ KOUNT,XN,HX,X
C
      Y=YA+HY(1)/2.
      DO 100 L=2,J
      L1=L-1
C
C OBTAIN U AT POINT MIDWAY BETWEEN TWO VERTICAL NET POINTS FOR THE
C PRESENT AND PREVIOUS STREAMWISE STATIONS
C
      DO 10 K=1,NU
      VH(K)=(UTX(K,L1)+UTX(K,L))/2.
   10 VHX(K)=(UT(K,L1)+UT(K,L))/2.
C
C OBTAIN W AT POINT MIDWAY BETWEEN TWO VERTICAL NET POINTS FOR THE
C PREVIOUS STREAMWISE STATION
C
      DO 20 K=1,NW
   20 UH(K)=(WTX(K,L1)+WTX(K,L))/2.
      CALL PREPP(Y)
C
C COMPUTE CONTRIBUTION FROM PREVIOUS STREAMWISE STATION FOR THE
C SPANWISE MOMENTUM EQUATION
C
      G(3,L)=G(1,L)*(WTX(2,L)+P3*WTX(1,L))-G(1,L1)*(WTX(2,L1)+P3*WTX(1,
     1  L1))+HY(L1)*((1.+P1)*VH(1)*UH(2)+4.*P4*VH(2)*UH(1)+4.*P6*
     2    (UH(1)*(VH(2)+VHX(2))-UH(2)*(VH(1)-VHX(1)))-SQRT(X)*P5*
     3    (VH(2)**2+VHX(2)**2))
      Y=Y+(HY(L1)+HY(L))/2.
  100 CONTINUE
C
      RETURN
      END
      SUBROUTINE BCW
C
C THIS SUBROUTINE COMPUTES THE JACOBIAN MATRIX AND RIGHT HAND SIDE
C FOR THE BOUNDARY CONDITION OF THE SMALL CROSS FLOW EQUAITON
C
      COMMON /SETUP/ UA(4),UB(4),G(4),B(4,4)
      DO 10 L=1,2
   10 G(L)=0.
      B(1,1)=1.
      B(2,1)=1.
      RETURN
      END
      SUBROUTINE RHSFW
C
C THSI SUBROUTINE COMPUTES THE RIGHT HAND SIDE OF THE SPANWISE MOMENTUM
C EQUATION
C
      COMMON /SETUP/ V(4),VX(4),F(4),A(4,4)
```

```
      DO 10 L=1,2
   10 F(L)=0.
      RETURN
      END
      SUBROUTINE JACOBW
C
C THIS SUBROUTINE COMPUTES THE JACOBIAN MATRIX FOR THE SMALL CROSS-FLOW
C STATION
C
      COMMON /PARM2/ P1,P2,P3,P4,P5,P6
      COMMON /SETUP/ V(4),VX(4),F(4),A(4,4)
      A(1,2)=1.
      A(2,1)=-4.*P4*V(2)+4.*P6*(V(2)+VX(2))
      A(2,2)=-(1.+P1)*V(1)-4.*P6*(V(1)-VX( ))
      RETURN
      END
      SUBROUTINE OUTPT(J,YA)
C
C THIS SUBROUTINE WRITES THE SOLUTION ON PAPER
C
      LOGICAL CASEU,CASEW
      COMMON /WI/ WI(2,151)
      COMMON /SOLVE/ CASEU,CASEW
      COMMON /UT/ UT(4,151) /MESHY/ H(151)
      Y=YA
      IF(CASEW) GO TO 200
C
C PRINT STREAMWISE VELOCITY VECTOR ONLY
C
      WRITE(6,6100)
      DO 110 L=1,J
      WRITE(6,6200) L,Y,(UT(K,L),K=1,4)
      Y=Y+H(L)
  110 CONTINUE
      RETURN
C
C PRINT BOTH STREMWISE AND SPANWISE VELOCITY VECTORS
C
  200 WRITE(6,6300)
      DO 300 L=1,J
      WRITE(6,6400) L,Y,(UT(K,L),K=1,4),(WI(K,L),K=1,2)
      Y=Y+H(L)
  300 CONTINUE
C
 6100 FORMAT(//10X,*Y*,14X,*F*,14X,*DF*,14X,*DDF*,13X,*P*//)
 6200 FORMAT(I4,5(1X,E14.7))
 6300 FORMAT(//10X,*Y*,14X,*F*,14X,*DF*,14X,*DDF*,13X,*P*,15X,*W*,15X,
     1   *DW*//)
 6400 FORMAT(I4,7(1X,E14.7))
C
      RETURN
```

```
      SUBROUTINE NETRON
C
C NET SELECTION - APPROXIMATELY CHOOSING H(K) WITH THE TRUNCATION ERROR
C TO BE A CONSTANT ON THE WHOLE INTERVAL
C
      COMMON /DU/ US(4,1) /UT/ UT(4,1) /HNEW/ HNEW(1) /MESHY/ H(1)
      COMMON /F/ F(4,1)
      COMMON /NET/ JMAX,HMAX,IFMX,KSAME
      COMMON /PARM4/ N,NW,J
      COMMON /PARM5/ FACX,HKS,XA,XB,YA,YB,NX
      COMMON /NET1/ KTYPE,KSINGK,KSING(1),KL,P(6),Q(6),R(6),TAU2(6)
      COMMON /SETUP/ UH(4),SUM(4),FF(4),AJA(4,4)
      DIMENSION Z(2000),KT(10)
      LOGICAL DEL1,DEL2
C
      KSINGK=1
      KSING(1)=J
      KEXCED=0
      NETINC=0
      AREA=0.
      KSAME=0
      IFMX=1
      J1=J-1
      DO 10 K=1,N
      SUM(K)=0.
   10 CONTINUE
C
C COMPUTE LOCAL TRUNCATION ERROR AT MID-POINT
C
      KTYPE=-1
      DO 300 L=1,J1
      KL=L
      IF(L.EQ.1) GO TO 100
      KTYPE=0
      IF(L.EQ.KSING(KSINGK)) GO TO 50
      IF(L.NE.(KSING(KSINGK)-1)) GO TO 100
C
C POINT BEFORE SINGULARITY OR RIGHT END-POINT
C
      KTYPE=1
      GO TO 100
   50 CONTINUE
C
C POINT AFTER SINGULARITY
C
      KTYPE=-1
      KSINGK=KSINGK+1
  100 CONTINUE
      DO 150 K=1,N
      UH(K)=(UT(K,L)+UT(K,L+1))/2.
```

```
  150 CONTINUE
      CALL THRON
      T=0.
      DO 200 K=1,N
      F(K,L)=TAU2(K)
      T=T+TAU2(K)**2
  200 CONTINUE
      T=SQRT(T)
      Z(L)=T
      AREA=AREA+H(L)**2*T/2.
  300 CONTINUE
C
      CONSTC=AREA/JI
C
C NET SELECTION
C
      LA=1
      DO 2000 KOUT=1,KSINGK
      LB=KSING(KOUT)-1
      IF(KOUT.GT.1) LA=KSING(KOUT)
      DEL1=.FALSE.
      DEL2=.FALSE.
      DO 1000 L=LA,LB
      FACT=H(L)**2*Z(L)/(2.*CONSTC)
      FACT=SQRT(FACT)
      Z(L)=FACT
      KJS=FACT+0.5
      IF(KJS.LE.1) GO TO 700
C
C ADDITION
C
      IF((L+KJS+NETINC+1).GT.JMAX) GO TO 2500
      KSAME=1
      IFMX=MAX0(IFMX,KJS)
      HI=H(L)/KJS
      DO 600 M=1,KJS
      HNEW(NETINC+M-1+L)=HI
      DO 600 K=1,N
      US(K,NETINC+M-1+L)=((M-1)*UT(K,L+1)+(KJS-(M-1))*UT(K,L))/KJS
  600 CONTINUE
      NETINC=NETINC+KJS-1
      DEL1=.FALSE.
      DEL2=.FALSE.
      GO TO 1000
  700 CONTINUE
C
      IF(FACT.GE.0.5) GO TO 800
      DEL2=.TRUE.
      IF(DEL1) GO TO 900
  800 CONTINUE
C
```

```
C
      IF((L+NETINC+1).GT.JMAX) GO TO 2500
      DEL1=DEL2
      DO 850 K=1,N
      US(K,L+NETINC)=UI(K,L)
  850 CONTINUE
      HNEW(L+NETINC)=H(L)
      GO TO 1000
  900 CONTINUE
C
C DELETION
C
      DEL1=DEL2
      IF((HNEW(L+NETINC-1)+H(L)).LE.HMAX) GO TO 825
      KEXCED=1
      GO TO 800
  825 CONTINUE
      KSAME=1
      HNEW(L+NETINC-1)=HNEW(NETINC+L-1)+H(L)
      NETINC=NETINC-1
 1000 CONTINUE
      KT(KOUT)=KSING(KOUT)+NETINC
 2000 CONTINUE
      DO 2100 K=1,KSINGK
      KSING(K)=KT(K)
 2100 CONTINUE
      IF(KSAME.EQ.0) RETURN
C
      JNEW=J+NETINC
      DO 2200 K=1,N
      US(K,JNEW)=UI(K,J)
 2200 CONTINUE
      J=JNEW
      HNEW(J)=0.
      DO 2300 L=1,J
      H(L)=HNEW(L)
      DO 300 K=1,N
      UI(K,L)=US(K,L)
 2300 CONTINUE
      RETURN
C
 2500 CONTINUE
      KSAME=0
C
      RETURN
      END
      SUBROUTINE TRUN
C
C THIS SUBROUTINE COMPUTES THE LOCAL TRUNCATION ERROR OF THE CENTERED-E
C SCHEME. THE INTEGER VARIABLE ** TYPE ** INDICATES THE FOLLOWING
```

```
:                 1   LEFT BOUNDARY POINT
:            =    0   INTERNAL POINT
:            =    1   RIGHT BOUNDARY POINT
: SINGULAR POINTS ARE TREATED AS BOUNDARY POINTS
:
      INTEGER TYPE
      COMMON /MESHY/ H(1) /UT/ UT(4,1)
      COMMON /PARM4/ N,NW,J
      COMMON /PARM5/ FACX,HKS,XA,XB,YA,YB,NA
      COMMON /NETI/ TYPE,KSINGK,KSING(1),L,P(6),Q(6),R(6),T(6)
      COMMON /SETUP/ UH(4),UHX(4),FF(4),A( ,4)
:
      IF(TYPE) 100, 200, 300
  100 CONTINUE
:
: LEFT BOUNDARY POINT
:
      A1=-H(L)/2.
      A2=-A1
      A3=A2+H(L+1)
      A4=A3+H(L+2)
      A5=A4+H(L+3)

      L1=L
      L2=L+1
      L3=L+2
      L4=L+3
      L5=L+4

      GO TO 400
  200 CONTINUE

  INTERNAL POINT

      IF(L.EQ.(KSING(KSINGK)-2)) GO TO 250

      A1=-H(L-1)-H(L)/2.
      A2=-H(L)/2.
      A3=H(L)/2.
      A4=A3+H(L+1)
      A5=A4+H(L+2)
:
      L1=L-1
      L2=L
      L3=L+1
      L4=L+2
      L5=L+3
:
      GO TO 400
  250 CONTINUE
:
```

```
C                                                                       00015680
      A1=H(L)/2.+H(L+1)                                                 00015690
      A2=H(L)/2.                                                        00015700
      A3=-A2                                                            00015710
      A4=A3-H(L-1)                                                      00015720
      A5=A4-H(L-2)                                                      00015730
C                                                                       00015740
      L1=L+1                                                            00015750
      L2=L                                                              00015760
      L3=L-1                                                            00015770
      L4=L-2                                                            00015780
      L5=L-3                                                            00015790
C                                                                       00015800
      GO TO 400                                                         00015810
  300 CONTINUE                                                          00015820
C                                                                       00015830
C RIGHT BOUNDARY POINT                                                  00015840
C                                                                       00015850
      A1=H(L)/2.                                                        00015860
      A2=-A1                                                            00015870
      A3=A2-H(L-1)                                                      00015880
      A4=A3-H(L-2)                                                      00015890
      A5=A4-H(L-3)                                                      00015900
C                                                                       00015910
      L1=L+1                                                            00015920
      L2=L                                                              00015930
      L3=L-1                                                            00015940
      L4=L-2                                                            00015950
      L5=L-3                                                            00015960
C                                                                       00015970
  400 CONTINUE                                                          00015980
C                                                                       00015990
C COMPUTE LOCAL TRUNCATION ERROR IN TWO STEPS                           00016000
C (A) CONTRIBUTION FROM THIRD DERIVATIVE                                00016010
C                                                                       00016020
      UP=VANDET(5,A1,A2,A3,A4,A5)                                       00016030
      C1=6.*(VANDET(3,A3,A4,A5,0.,0.)*A2**4                             00016040
     1    -VANDET(3,A2,A4,A5,0.,0.)*A3**4                               00016042
     2    +VANDET(3,A2,A3,A5,0.,0.)*A4**4                               00016044
     3    -VANDET(3,A2,A3,A4,0.,0.)*A5**4)/UP                           00016046
      C2=-6.*(VANDET(3,A3,A4,A5,0.,0.)*A1**4                            00016050
     1    -VANDET(3,A1,A4,A5,0.,0.)*A3**4                               00016052
     2    +VANDET(3,A1,A3,A5,0.,0.)*A4**4                               00016054
     3    -VANDET(3,A1,A3,A4,0.,0.)*A5**4)/UP                           00016056
      C3=6.*(VANDET(3,A2,A4,A5,0.,0.)*A1**4                             00016060
     1    -VANDET(3,A1,A4,A5,0.,0.)*A2**4                               00016062
     2    +VANDET(3,A1,A2,A5,0.,0.)*A4**4                               00016064
     3    -VANDET(3,A1,A2,A4,0.,0.)*A5**4)/UP                           00016066
      C4=-6.*(VANDET(3,A2,A3,A5,0.,0.)*A1**4                            00016070
     1    -VANDET(3,A1,A3,A5,0.,0.)*A2**4                               00016072
```

```
     2    +VANDET(3,A1,A2,A4,0.,0.)*A3**4                                00016074
     3    -VANDET(3,A1,A2,A3,0.,0.)*A5**4)/UP                            00016076
      C5=6.*(VANDET(3,A2,A3,A4,0.,0.)*A1**4                              00016080
     1    -VANDET(3,A1,A3,A4,0.,0.)*A2**4                                00016082
     2    +VANDET(3,A1,A2,A4,0.,0.)*A3**4                                00016084
     3    -VANDET(3,A1,A2,A3,0.,0.)*A4**4)/UP                            00016086
      DO 500 K=1,N                                                       00016090
      P(K)=C1*UT(K,L1)+C2*UT(K,L2)+C3*UT(K,L3)+C4*UT(K,L4)+C5*UT(K,L5)  00016100
  500 CONTINUE                                                           00016110
C                                                                        00016120
C (B) CONTRIBUTION FROM SECOND DERIVATIVE                                00016130
C                                                                        00016140
      UP=VANDET(4,A1,A2,A3,A4,0.)                                        00016150
      C1=2.*(VANDET(2,A3,A4,0.,0.,0.)*A2**3                              00016160
     1    -VANDET(2,A2,A4,0.,0.,0.)*A3**3                                00016162
     2    +VANDET(2,A2,A3,0.,0.,0.)*A4**3)/UP                            00016164
      C2=-2.*(VANDET(2,A3,A4,0.,0.,0.)*A1**3                             00016170
     1    -VANDET(2,A1,A4,0.,0.,0.)*A3**3                                00016172
     2    +VANDET(2,A1,A3,0.,0.,0.)*A4**3)/UP                            00016174
      C3=2.*(VANDET(2,A2,A4,0.,0.,0.)*A1**3                              00016180
     1    -VANDET(2,A1,A4,0.,0.,0.)*A2**3                                00016182
     2    +VANDET(2,A1,A2,0.,0.,0.)*A4**3)/UP                            00016184
      C4=-2.*(VANDET(2,A2,A3,0.,0.,0.)*A1**3                             00016190
     1    -VANDET(2,A1,A3,0.,0.,0.)*A2**3                                00016192
     2    +VANDET(2,A1,A2,0.,0.,0.)*A3**3)/UP                            00016194
C                                                                        00016200
      DO 600 K=1,N                                                       00016210
      R(K)=C1*UT(K,L1)+C2*UT(K,L2)+C3*UT(K,L3)+C4*UT(K,L4)               00016220
  600 CONTINUE                                                           00016230
      CALL JACOB                                                         00016240
C                                                                        00016250
      DO 800 K=1,N                                                       00016260
      Q(K)=0.                                                            00016270
      DO 700 M=1,N                                                       00016280
      Q(K)=Q(K)+A(K,M)*R(M)                                              00016290
  700 CONTINUE                                                           00016300
  800 CONTINUE                                                           00016310
C                                                                        00016320
C SECOND ORDER TRUNCATION ERROR                                          00016330
C                                                                        00016340
      DO 900 K=1,N                                                       00016350
      T(K)=P(K)-3.*Q(K)                                                  00016360
  900 CONTINUE                                                           00016370
C                                                                        00016380
      RETURN                                                             00016390
      END                                                                00016400
      FUNCTION VANDET(N,X1,X2,X3,X4,X5)                                  00016410
C                                                                        00016420
C THIS FUNCTION ROUTINE COMPUTES THE DETERMINANT OF AN (N * N)           00016430
C VANDERMONDE MATRIX WITH ENTRIES X(1),X(2),......, WITH N GREATER THAN  00016440
C 1 AND LESS THAN 7                                                      00016450
```

```
      DIMENSION X(5)
C
      X(1)=X1
      X(2)=X2
      X(3)=X3
      X(4)=X4
      X(5)=X5
      N1=N-1
      VANDET=1.
      DO 200 L=1,N1
      L1=L+1
      DO 100 K=L1,N
      VANDET=VANDET*(X(K)-X(L))
  100 CONTINUE
  200 CONTINUE
C
C
      RETURN
      END
      SUBROUTINE YMESH
C
C SUBROUTINE IN WHICH USER SUPPLIES OWN VERTICAL MESH AND TOTAL
C NUMBER OF NET POINTS = (NUMBER OF INTERNAL INTERVALS +1)
C BY VERTICAL MESH, WE MEAN THE VECTOR H(L), WHERE Y(L+1)=Y(L)+H(L), FOR
C L GREATER THAN ZERO, WITH Y(1)=YA, Y(J)=YB
C
      COMMON /PARM4/ N,NW,J
      COMMON /MESHY/ HY(1)
C
      RETURN
      END
      SUBROUTINE XMESH
C
C SUBROUTINE IN WHICH USER SUPPLIES OWN STREAMWISE MESH AND TOTAL
C NUMBER OF STREAMWISE STATIONS TO BE MARCHED (THIS EXCLUDE THE STATION
C AT S=0). BY STREAMWISE MESH, WE MEAN THE VECTOR HX(L), WHERE X(L+1) =
C X(L) +HX(L), FOR L GREATER THAN ZERO, WITH X(1)=XA AND X(NX+1)=XB
C
      COMMON /MESHX/ HX(1)
      COMMON /PARM5/ FACX,HKS,XA,XB,YA,YB,NX
C
      RETURN
      END
      SUBROUTINE PRMESH
      COMMON /KPRT/ KPRT(1)
C
C SUBROUTINE IN WHICH USER SUPPLIES OWN STREAMWISE STATIONS AT WHICH
C SOLUTION WILL BE PRINTED ON PAPER. THE VECTOR KPRT(L) SHOULD HAVE
C THE PROPERTY KPRT(L) IS GREATER THAN KPRT(K), FOR L GREATER THAN K.
C KPRT(M)=1 MEANS SOLUTIONS HAVE BEEN PRINTED FOR (M-1) STATIONS. (THE
```

```
C M-TH STATION TO HAVE THE SOLUTION PRINTED. NOTE THAT M IS STRICTLY
C LESS THAN I
C
      RETURN
      END
```

APPENDIX B: 3-D WALL-JET SMALL CROSS FLOW PROGRAM--RUNNING INSTRUCTIONS

The following deck setup will indicate control cards and data cards (a name list card INPUTS) needed to run the following example:

(A) Small cross flow CASEW=.TRUE.

(B) u = 0 at outer edge of jet, C4 = 0.

(C) Parameter in logarithmic spiral K = 1/3, C1 = 1./3.

(D) The induced magnitude of induced cross-flow K_2 = 1, C7 = -1.

(E) Meshes, initial profile + streamwise output stations are all to be provided by program.

(F) Streamwise station to be solved to XB = 1.

(G) Initial mass flux = 1.

1. <u>Deck Setup</u>

Job Card

FTN4.

LGO.

7 / 8 / 9

Source program

7 / 8 / 9

-$ INPUTS CASEW=.TRUE., C1=.33333333, C4=0., C7=-1.$

↑
Second column

6 / 7 / 8 / 9

For other desired cases, see definitions of the various variables and their options in the listing. The solutions are to be printed on paper

in sequence of streamwise stations. Thus, for any s location, there are eight columns of outputs:

Column 1: index of vertical net points

Column 2: y--vertical net point values, from y = 0 (1 in column 1, at the wall) to y = YB (the last value in column 1, outer edge of jet)

The next six columns have the same convention as column 2:

Column 3: F--Glauert similarity variable

Column 4: DF--the partial derivative ∂/∂(eta) F

Column 5: DDF--the partial derivative ∂/∂(eta) DF

Column 6: P--the reduced pressure P

Column 7: W--cross-flow velocity

Column 8: DW--the partial derivative ∂/∂(eta) W

To find out the computed values of (F, DF, DDF, P, W, DW) at y = 0.5, say, we need to look at the horizontal line with the y-value on column 2 to match with y = 0.5 (provided y = 0.5 is a net point), then the third to eighth columns will give the function values of F, DF, DDF, P, W, DW at y = 0.5.

2. Type and Configuration of Computer Used in Program Development

   (i) Lawrence Berkeley Laboratory 7600

   (ii) CDC 6600 at Arbor Vitae, Los Angeles, and Sunnyvale

3. Estimate of Running Time

   16 seconds on CDC 7600

4. Name and Level of Programming Language Used in Program

   FORTRAN IV